Hello,

I tried compiling ruby 3.1 preview on OpenIndiana (Illumos) and this works:

```
./configure --prefix=$HOME/ruby-31 --with-gcc --disable-dtrace
gmake
```

```
gmake install
```

It works:

```
$HOME/ruby-31/bin/ruby --version
ruby 3.1.0preview1 (2021-11-09 master 5a3b2e6141) [i386-solaris2.11]
```

It successfully compiles with the above config line and seems to work.

You can see in the above that it builds a 32bit by default, I did not yet try a full 64bit compile.

Note that I use the --with-gcc to select the OpenIndiana GCC compiler (not OpenIndiana clang).

I compiled with the default gcc 7.5.

Now for some reason it seems I have to use --disable-dtrace.

In older versions < 3.0 this works with DTRACE enabled, but this is apparently broken since 3.0.x

If I compile with dtrace enabled I get:

```
checking whether dtrace USDT is available... yes(-xnolibs)
checking whether dtrace needs post processing... rebuild
```

and then during compile

```
compiling builtin.c
```

```
generating a glommed object with DTrace probes for static library
```
```
linking static-library libruby-static.a
```
```
gar: ruby-glommed.o: SHT_GROUP section [index 52] has no SHF_GROUP sections
```
```
gar: ruby-glommed.o: SHT_GROUP section [index 53] has no SHF_GROUP sections
```
```
gar: ruby-glommed.o: SHT_GROUP section [index 54] has no SHF_GROUP sections
```
```
gar: ruby-glommed.o: SHT_GROUP section [index 55] has no SHF_GROUP sections
```
```
... lots of similar messages
```
```
gar: ruby-glommed.o: unknown type [0] section ' in group [group]
gar: ruby-glommed.o: unknown type [0] section' in group [group]
gar: ruby-glommed.o: unknown type [0] section `` in group [group]
```

Does anyone know what ruby-glommed is and what exactly broke here?

It seems some sort of rebuild code generation is done in the dtrace case, but I am not familiar with the ruby config/build system to see what exactly.

The problem seems to be with GNU ar (gar).

There exist similar reports for FreeBSD like

https://bugs.ruby-lang.org/projects/ruby-master/repository/git/commits/95aff214687a5e12c3eb57d056665741e734c188
"The latest ruby cannot compile with FreeBSD Dtrace enabled."

Is there a way please to compile with dtrace enabled?

Thank you,
David Stes

---

**History**

#1 - 11/16/2021 02:45 PM - stes (David Stes)
- Assignee changed from ngoto (Naohisa Goto) to stes (David Stes)

#2 - 11/16/2021 02:46 PM - stes (David Stes)
- Assignee deleted (stes (David Stes))

#3 - 11/16/2021 06:15 PM - stes (David Stes)
The problem may be due to the presence on OpenIndiana of the Illumos /usr/bin/ar and the GNU /usr/bin/gar. The configure script selects "gar".

When configuring like this:

```
./configure --prefix=$HOME/ruby-31 --with-gcc AR=/usr/bin/ar
```

then the "gar" errors disappear and it seems to do correctly:

compiling builtin.c
generating a glommed object with DTrace probes for static library
linking static-library libruby-static.a

#4 - 11/18/2021 06:43 PM - stes (David Stes)
The actual configure line that works for me on OpenIndiana 2021.10 with gcc 10.3 (or gcc 7.5) is:

```
./configure --prefix=$HOME/ruby-31 --with-gcc --enable-dtrace CFLAGS="-g -O2" AR=/usr/bin/ar STRIP=/usr/bin/strip
```

Note that I compile with gcc either 7.5 (32bit) or 10.3 (64bit).

I use --enable-dtrace which uses something called a "glommed object" as discussed in

[https://marc.info/?l=opensolaris-dtrace-discuss&m=114761203110734&w=4](https://marc.info/?l=opensolaris-dtrace-discuss&m=114761203110734&w=4)

which is a link that is in the Ruby Makefile to explain the idea behind the glommed object.

Finally I have to add CFLAGS="-g -O2" to the configure line.

If not, then it uses the -O3 flag which is set in the configure.ac script as optflags, but this crashes 'ld' (signal 11) in a text in bigdecimal as can be seen in the ext/bigdecimal/mkmf.log

The ruby-2.7.4 configuration is easier since (1) it does not use GNU ar (gar) and (2) the ruby 2.7.4 configure script sets CFLAGS to -g -O2 and optflags to -O3 and it seems it is not setting both flags to -O3.

#5 - 11/19/2021 02:02 PM - ngoto (Naohisa Goto)
- Status changed from Open to Third Party's Issue

Thanks for reporting.
I suspect that OpenIndiana's ld and GNU ar (gar) have some problems for treating dtrace binaries.

I thought optflags could be specified optimization flags.

```
./configure --prefix=$HOME/ruby-31 --with-gcc --enable-dtrace optflags="-O2" AR=/usr/bin/ar STRIP=/usr/bin/strip
```

#6 - 11/20/2021 02:46 PM - stes (David Stes)
Thank you for your response.

The good news is that --enable-dtrace still works with ruby 3.1 (and 3.0.3).

I compared with ruby 2.7.4 and ruby 2.7.4 sets CFLAGS to "-g -O2" and optflags to -O3.

These flags are different, it seems.
Effectively sometimes -O3 is used and sometimes -g -O2 is used.
By using this old behavior in the 3.0.2 and 3.0.3 and 3.1 builds, I get a succesful build;

Based on the info in https://marc.info/?l=opensolaris-dtrace-discuss&m=114761203110734&w=4 I also tried adding --enable-shared to the configure line when I add --enable-dtrace.

However for me --enable-shared does not help: this still gives rb_rational_num undefined in mkmf.log of ext/bigdecimal.

Anyway, by using the configure line as stated before I can run "gmake install" successfully and I have built thus ruby 3.1 and ruby 3.0.3 on OpenIndiana with gcc 7.5 and with gcc 10.3.

If GNU ar (gar) is somehow having an issue , what issue would that be exactly ?

If necessary I can log an issue about gar (GNU ar) with openindiana.

#7 - 11/25/2021 06:44 AM - nto (Naohisa Goto)
On Solaris 10 (sparc64-sun-solaris2.10), gcc 7.5 with gcc-ar works fine with dtrace.
So, the problem may be Illumos and/or OpenIndiana specific.
Please consult DTrace specialists in Illumos or OpenIndiana.

However for me --enable-shared does not help: this still gives rb_rational_num undefined in mkmf.log of ext/bigdecimal.

Generally, old version of libruby.a, libruby.so, and/or ruby.h in somewhere in your system may cause such kind of undefined symbol error.
Environment variables LD_LIBRARY_PATH (including LD_LIBRARY_PATH_32 or LD_LIBRARY_PATH_64), LD_RUN_PATH, LD_PRELOAD, etc. also might be a cause of linking unexpected libraries.

#8 - 11/28/2021 01:09 PM - stes (David Stes)
- Subject changed from openindiana ruby 3.1 preview needs --disable-dtrace to openindiana ruby 3.1 --enable-dtrace requires CFLAGS="-g"

I've changed the subject to "--enable-dtrace" requires CFLAGS="-g"

To be honest, what I tested so far is : with.gnu-binutils 2.36 (gar version 2.36) installed,
I can say that gar itself works, as long as I set CFLAGS="-g -O2".

However I suspect the -O2 is irrelevant, presumably "-O3" (which is the default in Ruby 3.1) also works.

However I think the -g flag adds debug info. It seems possible to me that the "dtrace" rebuild is scanning the symbol tables (perhaps debug info) to
generate code that support DTrace, and that the DTrace build depends on -g.

Ruby 2.7 has CFLAGS="-g -O2". This made me test those flags with Ruby 3.0 as well and Ruby 3.1.

The situation is certainly not bad, because I am just and simply reporting that Ruby 3.1 with DTrace enabled works fine for me on OpenIndiana, even
with gar 2.36, as long as I use CFLAGS="-g -O2" for the build which was the default in Ruby 2.7 but not in Ruby 3.x any longer.

Assuming the Ruby teams supports DTrace, this is also a Ruby build issue, it is not only an OpenIndiana issue.

Although that I agree that this issue can be closed.
It is OK by me to close the issue , thanks for your feedback, and perhaps my investigation can help somebody with a similar Ruby/DTrace issue.

Thanks,

#9 - 11/30/2021 11:03 AM - ngoto (Naohisa Goto)
- Status changed from Third Party's Issue to Rejected

By default, Ruby's configure script automatically sets -ggdb3 or -g to debugflags after checking if the compiler accepts one of the options. The
debugflags is included in CFLAGS by default.

However, setting CFLAGS completely wipes out the automatically set options by the configure script, for overriding CFLAGS by a user.
When setting CFLAGS , all contents of CFLAGS should be managed by the user who sets CFLAGS.

Instead of using CFLAGS , cflags is used for adding extra options to the C compiler.
For details, see ./configure --help.

If you mean that OpenIndiana's dtrace does not accept -ggdb3 and only accepts -g, setting debugflags="-g" may help.

#10 - 12/02/2021 07:58 AM - stes (David Stes)
- Subject changed from openindiana ruby 3.1 --enable-dtrace requires CFLAGS="-g" to openindiana ruby 3.1 --enable-dtrace requires debugflags="-g!"
Thanks for explaining. Indeed, I tested and can confirm that, by default it seems to set debugflags to `-g3` (or `-ggdb3` actually) but I think `-ggdb3` is equivalent to `-g3`. So by default it uses level 3 debug info `-ggdb3`. For the --enable-dtrace build, where extra DTrace related operations are being done, it seems the build fails when compiling with `-ggdb3` but it works when setting debugflags=`-g1`.

So the following config line works for me with DTrace and gnu-binutils (gar) 2.36 and gcc 10.3:

```
./configure --with-gcc --enable-dtrace debugflags="-g1"
```

Would it be possible to change the default `-g3` to `-g1`?

My investigation in what is different with Ruby 2.7 (which works with DTrace enabled) indicates that Ruby 2.7 uses `-g1`.

#11 - 12/03/2021 12:10 PM - ngoto (Naohisa Goto)
- Status changed from Rejected to Third Party’s Issue

OpenIndiana’s gcc + gnu binutils + dtrace only work with `-g1` seems to be OpenIndiana specific problem and this should essentially be resolved in OpenIndiana.
As far as I can see, such problem is not reported in other platforms.

According to GCC's documentation, `-ggdb3` is not equivalent to `-g3`.
https://gcc.gnu.org/onlinedocs/gcc/Debugging-Options.html#Debugging-Options
The document also says `-g` is equivalent to `-g2`, not `-g1`.

#12 - 12/04/2021 03:33 PM - stes (David Stes)

Thanks, but I think this is partly a Ruby configuration issue.

The goal of "configure" is to find suitable configuration for the target operating system so that the build works on the target OS where configure is ran.

The Ruby configure script could be enhanced with a test to enable `-ggdb3` only if (1) the compiler accepts it and (2) the other tools like the link editor can deal with it.

If such a configure script test can be implemented, this would also help me to log a bug report with OpenIndiana/Illumos to indicate that they have a bug.

If the configure script can detect that it should not enable `-ggdb3` on OpenIndiana, then the test in the configure script can also see that it is OK to enable `-ggdb3` on other operating systems.

#13 - 12/08/2021 05:15 PM - ngoto (Naohisa Goto)

Patches are welcome to check them without breaking things on other platforms (including Oracle Solaris 11).

I guess that the error/failure only occurs when processing large-scale complex source codes, and simple short source codes could be processed without error/failure. In such cases, I think it is very difficult to write short check codes that can be used in configure scripts, and I give up trying to do so.