Ruby master - Bug #5679

Too many arguments for format warnings on mingw32 build

11/28/2011 02:28 PM - phasis68 (Heesob Park)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>luislavena (Luis Lavena)</td>
</tr>
<tr>
<td>Target version:</td>
<td>2.0.0</td>
</tr>
<tr>
<td>ruby -v:</td>
<td>-</td>
</tr>
</tbody>
</table>

Description

I can see the following warnings during mingw32 build.

gc.c: In function 'gc_profile_result':
gc.c:3581:4: warning: unknown conversion type character 'z' in format
gc.c:3581:4: warning: unknown conversion type character 'z' in format
gc.c:3581:4: warning: unknown conversion type character 'z' in format
gc.c:3581:4: warning: format '%30.20f' expects type 'double', but argument 5 has type 'unsigned int'
gc.c:3581:4: warning: too many arguments for format

transcode.c: In function 'str_transcode0':
transcode.c:2734:9: warning: unknown conversion type character 't' in format
transcode.c:2734:9: warning: too many arguments for format

iseq.c: In function 'insn_operand_intern':
iseq.c:826:50: warning: unknown conversion type character 't' in format
iseq.c:826:50: warning: too many arguments for format
iseq.c: In function 'rb_iseq_disasm_insn':
iseq.c:860:2: warning: unknown conversion type character 'z' in format
iseq.c:860:2: warning: format '%-16s' expects type 'char *', but argument 3 has type 'size_t'
iseq.c:860:2: warning: too many arguments for format
iseq.c:864:7: warning: unknown conversion type character 'z' in format
iseq.c:864:7: warning: format '%-16.*s' expects type 'char *', but argument 4 has type 'int'
iseq.c:864:7: warning: too many arguments for format

In file included from vm_exec.c:106:0,
from vm.c:25:
insns.def: In function 'vm_exec_core':
insns.def:1087:6: warning: unknown conversion type character 't' in format
insns.def:1087:6: warning: unknown conversion type character 't' in format
insns.def:1087:6: warning: too many arguments for format

rubyext.c: In function 'rb_syck_err_handler':
rubyext.c:687:12: warning: unknown conversion type character 't' in format
rubyext.c:687:12: warning: format '%s' expects type 'char *', but argument 5 has type 'int'
rubyext.c:687:12: warning: too many arguments for format

And I know these warnings are completely meaningless because Ruby uses it's own BSD_vfprintf implementaion.

Here is a patch of include/ruby/ruby.h for suppressing format warnings on mingw32

--- ruby.h 2011-11-28 14:21:11.000000000 +0900
+++ ruby.h.new 2011-11-28 14:23:01.000000000 +0900
@@ -37,7 +37,7 @@
    # define NOINLINE(x) x
@end

-#ifdef GNUC
+#elif defined(GNUC) && !defined(MINGW32)
define PRINTF_ARGS(decl, string_index, first_to_check) \
defl attribute((format(printf, string_index, first_to_check)))
#else

09/14/2021 1/4
I can see the following warnings during mingw32 build.

gc.c: In function 'gc_profile_result':
  gc.c:3581:4: warning: unknown conversion type character 'z' in format
gc.c:3581:4: warning: unknown conversion type character 'z' in format
gc.c:3581:4: warning: unknown conversion type character 'z' in format
gc.c:3581:4: warning: format '%30.20f' expects type 'double', but argument 5 has type 'unsigned int'
gc.c:3581:4: warning: too many arguments for format

Where does 'z' come from?
And I know these warnings are completely meaningless because Ruby uses its own BSD_vfprintf implementation.

Here is a patch of include/ruby/ruby.h for suppressing format warnings on mingw32

Your analysis doesn't have why mingw makes such annoying warnings. And, in fact, 99% printf don't make such annoying warnings. So, it's no good idea blindly suppressing.

#2 - 11/29/2011 12:23 AM - luislavena (Luis Lavena)
On Mon, Nov 28, 2011 at 11:56 AM, KOSAKI Motohiro kosaki.motohiro@gmail.com wrote:

I can see the following warnings during mingw32 build.

```
gc.c: In function 'gc_profile_result':
gc.c:3581:4: warning: unknown conversion type character 'z' in format
gc.c:3581:4: warning: unknown conversion type character 'z' in format
gc.c:3581:4: warning: unknown conversion type character 'z' in format
gc.c:3581:4: warning: format '%30.20f' expects type 'double', but argument 5 has type 'unsigned int'
gc.c:3581:4: warning: too many arguments for format
```

Where does 'z' come from?

PRI?SIZE from ruby.h

Anyhow, I think this is coming from printf implementation of MinGW:

http://gcc.gnu.org/bugzilla/show_bug.cgi?id=7768

And this:


Quoting:

"On 05.12.2010 6:36, Simson Garfinkel wrote:

Hi. Another porting question.

GCC on every platform I use has a %z format for size_t data. I even saw that on mingw with gcc 3.5 when run as a cross-compiler. But the %z format seems missing from the 4.x gcc running on Windows-hosted GCC. What should I do?

bloom.c:221:5: warning: unknown conversion type character 'z' in format"

CFLAGS

#3 - 11/29/2011 12:23 AM - kosaki (Motohiro KOSAKI)

Anyhow, I think this is coming from printf implementation of MinGW:

http://gcc.gnu.org/bugzilla/show_bug.cgi?id=37768

And this:


Quoting:

"On 05.12.2010 6:36, Simson Garfinkel wrote:

Hi. Another porting question.

GCC on every platform I use has a %z format for size_t data. I even saw that on mingw with gcc 3.5 when run as a cross-compiler. But the %z format seems missing from the 4.x gcc running on Windows-hosted GCC. What should I do?

bloom.c:221:5: warning: unknown conversion type character 'z' in format"
Great analysis!

So, I'm ok both Heesob's patch (but please add some comments) and CFLAGS change.

thank you.

#4 - 12/08/2011 07:59 AM - kosaki (Motohiro KOSAKI)
- Category set to core
- Status changed from Open to Assigned
- Assignee set to luislavena (Luis Lavena)
- Target version set to 2.0.0

#5 - 12/08/2011 12:11 PM - nobu (Nobuyoshi Nakada)
- Status changed from Assigned to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r33978.
Heesob, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- configure.in (RUBY_WERROR_FLAG): append all warning flags which are enabled to compile, so that printf format modifiers properly fail.
  [ruby-core:41351] [Bug #5679]