Forced definition of struct timezone under MinGW

01/10/2010 06:07 PM - luislavena (Luis Lavena)

Status: Closed
Priority: Normal
Assignee:
Target version: 2.0.0
ruby -v: ruby 1.9.1p376 (2009-12-07 revision 26041) [i386-mingw32]

Description

It appears that win32.h is forcing the definition of timezone structure, even when it has been defined by MinGW provided headers.

The output from compilation of miniruby:

In file included from ../../../include/ruby/defines.h:205:0,
from ../../../include/ruby/ruby.h:73,
from ../../../include/ruby.h:32,
from ../../../main.c:13:
../../../include/ruby/win32.h:201:8: error: redefinition of 'struct timezone'

And the code that defines it:

 ifndef MINGW32
 struct timezone {
 int tz_minuteswest;
 int tz_dsttime;
 };
 ifndef isascii
#define isascii __isascii
endif

It is clear that this was made to workaround limitations of older versions of MinGW, but a simple check with AC_CHECK_TYPES (or if specific are needed: AC_CHECK_MEMBERS), any of those could provide the defines needed to avoid the forced redefinition.

Please find attached a naive patch for configure.in and win32.h that use HAVE_STRUCT_TIMEZONE

Thank you.

Related issues:
Related to Backport187 - Backport #3666: Backport of r26311 (Bug #2587) Closed 08/08/2010

History

#1 - 01/12/2010 07:48 AM - rogerdpack (Roger Pack)

=begin
is this a cross compile? Which version of gcc?
r
=end

#2 - 01/12/2010 07:54 AM - luislavena (Luis Lavena)

=begin
Roger:

No cross-compiler, native.

Timezone is part of the newer w32api packages for MinGW. Will grab the versions and follow once I have the computer around.

--
This issue was solved with changeset r26311.
Luis, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>check-timezone-struct-avoid-redefinition.patch</td>
<td>956</td>
<td>01/10/2010</td>
<td>luislavena (Luis Lavena)</td>
</tr>
</tbody>
</table>