Ruby - Bug #7660

VC Builds Broken

01/06/2013 07:11 PM - cfis (Charlie Savage)

Status:	Rejected		
Priority:	Normal		
Assignee:	usa (Usaku NAKAMURA)		
Target version:	2.0.0		
ruby -v:	1.9.3 p362 and ruby trunk	**Backport:**	

Description

Ruby 1.9.3 p362 and Ruby 2.0.0 can no longer be build with VC 2010, while Ruby 1.9.3 p286 and earlier are fine. The problem is with win32/file, which is compiled twice while ruby/file is not compiled at all. Looks like something changed in the makefiles betweeen p286 and p362.

Here are the relevant parts of the log:

c:\MinGW\local\src\ruby\win32>nmake

Microsoft (R) Program Maintenance Utility Version 10.00.30319.01
Copyright (C) Microsoft Corporation. All rights reserved.

```
    CC = cl -nologo
    LD = cl -nologo
    LDSHARED = cl -nologo -LD
    CFLAGS = -MD -Zi -W2 -wd4996 -we4028 -we4142 -O2sy-  -Zm600
    XCFLAGS = -DRUBY_EXPORT -I. -I.ext/include/i386-mswin32_100 -I./../include -I./.. -I./../missi
ng
    CPPFLAGS =
    DLDFLAGS = -incremental:no -debug -opt:ref -opt:icf -dll
    SOLIBS =
```

Creating config.h
.ext\include\i386-mswin32_100\ruby\config.h unchanged.
Creating verconf.h
verconf.h unchanged.
Creating config.status
compiling ./../main.c
main.c

compiling file.c
file.c
compiling ./../gc.c
gc.c

compiling ./../win32/file.c
file.c

oldnames.lib user32.lib advapi32.lib shell32.lib ws2_32.lib imagehlp.lib shlwapi.lib
linking miniruby.exe
file.obj : error LNK2005: _rb_file_expand_path_internal already defined in file.obj
file.obj : error LNK2005: _rb_file_load_ok already defined in file.obj
file.obj : error LNK2005: _rb_w32_init_file already defined in file.obj
dir.obj : error LNK2019: unresolved external symbol _rb_str_encode_ospath referenced in function _dir_initialize
io.obj : error LNK2001: unresolved external symbol _rb_str_encode_ospath
ruby.obj : error LNK2001: unresolved external symbol _rb_str_encode_ospath
iseq.obj : error LNK2001: unresolved external symbol _rb_get_path
dir.obj : error LNK2019: unresolved external symbol _rb_get_path referenced in function _dir_initialize
load.obj : error LNK2001: unresolved external symbol _rb_get_path
io.obj : error LNK2001: unresolved external symbol _rb_get_path
process.obj : error LNK2001: unresolved external symbol _rb_get_path
dir.obj : error LNK2019: unresolved external symbol _rb_enc_path_end referenced in function _check_dirname
dir.obj : error LNK2019: unresolved external symbol _rb_enc_path_skip_prefix referenced in function _check_dirname

```
dir.obj : error LNK2019: unresolved external symbol _rb_get_path_no_checksafe referenced in function _file_s_fnmatch
dir.obj : error LNK2019: unresolved external symbol _rb_home_dir referenced in function _dir_s_home
dir.obj : error LNK2019: unresolved external symbol _rb_file_const referenced in function _Init_Dir
dir.obj : error LNK2001: unresolved external symbol _rb_cFile
io.obj : error LNK2001: unresolved external symbol _rb_cFile
dir.obj : error LNK2019: unresolved external symbol _rb_file_directory_p referenced in function _Init_Dir
dln_find.obj : error LNK2019: unresolved external symbol _eaccess referenced in function _dln_find_1
eval.obj : error LNK2019: unresolved external symbol _rb_file_dirname referenced in function _f_current_dirname
load.obj : error LNK2001: unresolved external symbol _rb_file_dirname
load.obj : error LNK2019: unresolved external symbol _rb_file_expand_path_fast referenced in function
_rb_construct_expanded_load_path
load.obj : error LNK2019: unresolved external symbol _rb_get_path_check_convert referenced in function
_rb_construct_expanded_load_path
load.obj : error LNK2019: unresolved external symbol _rb_is_absolute_path referenced in function
_rb_construct_expanded_load_path
load.obj : error LNK2019: unresolved external symbol _rb_get_path_check_to_string referenced in function
_rb_construct_expanded_load_path
load.obj : error LNK2019: unresolved external symbol _rb_realpath_internal referenced in function _rb_load_internal
ruby.obj : error LNK2001: unresolved external symbol _rb_realpath_internal
iseq.obj : error LNK2001: unresolved external symbol _rb_realpath_internal
load.obj : error LNK2019: unresolved external symbol _rb_find_file_ext_safe referenced in function _search_required
load.obj : error LNK2019: unresolved external symbol _rb_find_file_safe referenced in function _search_required
load.obj : error LNK2019: unresolved external symbol _rb_find_file referenced in function _rb_load
load.obj : error LNK2019: unresolved external symbol _rb_file_absolute_path referenced in function _rb_f_require_relative
hash.obj : error LNK2019: unresolved external symbol _rb_path_check referenced in function _path_tainted_p
io.obj : error LNK2019: unresolved external symbol _Init_File referenced in function _Init_IO
ruby.obj : error LNK2019: unresolved external symbol _rb_file_expand_path referenced in function _expand_include_path
miniruby.exe : fatal error LNK1120: 23 unresolved externals
NMAKE : fatal error U1077: '"C:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\BIN\cl.EXE"' : return code '0x2'
Stop.
```

History

#1 - 01/06/2013 07:15 PM - cfis (Charlie Savage)

Looks like the issue is these two rules:

```
{$(srcdir)}.c{}.obj:
$(ECHO) compiling $(<:=/)
$(Q) $(CC) $(CFLAGS) $(XCFLAGS) $(CPPFLAGS) $(COUTFLAG)$@ -c -Tc$(<:=/)
.c.obj:
$(ECHO) compiling $(<:=/)
$(Q) $(CC) $(CFLAGS) $(XCFLAGS) $(CPPFLAGS) $(COUTFLAG)$@ -c -Tc$(<:=/)
```

Instead of file.c being compiled by the first rules, its being compiled by the second one (note that the working directory for running nmake is ruby/win32). Flipping them solves the problem, but then looks like the wrong version of miniprelude is compiled (the ruby one, not ruby/win32 one).

#2 - 01/06/2013 07:30 PM - cfis (Charlie Savage)

Simply removing this rule:

```
.c.obj:
$(ECHO) compiling $(<:=/) $(Q) $(CC) $(CFLAGS) $(XCFLAGS) $(CPPFLAGS) $(COUTFLAG)$@ -c -Tc$(<:=/)
```

Solves the problem.

#3 - 01/07/2013 12:11 AM - luislavena (Luis Lavena)

- Category set to build

- Status changed from Open to Assigned

- Assignee set to usa (Usaku NAKAMURA)

#4 - 01/19/2013 11:17 AM - cfis (Charlie Savage)

Any updates on this?

#5 - 01/21/2013 04:36 PM - usa (Usaku NAKAMURA)

- Status changed from Assigned to Rejected

=begin
at the bottom win32/README.w32 of trunk:
You can build ruby in any directory including the source directory,
except (({win32})) directory in the source directory.
Maybe I should backport this comment to 1.9.3 :P
=end

#6 - 01/21/2013 05:37 PM - cfis (Charlie Savage)

I see, missed that note. Since this is a recent change in 1.9.3 I think it should be backported.

However, my proposed allows building from the win32 directory. It also works when building from the top level directory. Do you think it breaks some cases? If not, then why not apply it?

Thanks - Charlie