

Backport193 - Backport #7174

Advocating for backporting 36811

10/16/2012 11:56 PM - jonforums (Jon Forums)

Status:	Closed
Priority:	Normal
Assignee:	usa (Usaku NAKAMURA)
Description	
Please consider backporting 36811 and refinements for the following reasons:	
1) While it's debatable whether this patch fixes a Windows platform bug in the 1.9.x series or whether it's a performance refactoring, it is a fact that this patch fixes long-standing 1.9.x usability problems important to many Ruby on Windows users. https://gist.github.com/3242245	
2) For many months Hiroshi has maintained a 1.9.3 backport of this functionality, with the latest version available at https://github.com/thecodeshop/ruby/tree/win-file-trunk/ruby_1_9_3 . This functionality has been part of multiple tcs-ruby experimental releases over the last 6-9 months.	
3) While the functionality has been tested on 1.9.3 via tcs-ruby by many interested devs over many months, tcs-ruby is not (and will never be) an alternative MRI distribution. tcs-ruby's sole purpose is to be collaboration repo and proving ground for those interested in contributing Windows and cross-platform improvements to MRI. tcs-ruby's goal is to refine the improvements and make them available for easy inclusion into MRI.	
4) As a consequence of (3), Hiroshi's 1.9.3 backports have not had as much widespread testing as they would have if they were included in an official RubyInstaller release. This was a purposeful decision on our part. RubyInstaller releases are meant to track official ruby-core releases. They are meant to be stable, not experimental releases.	
As this fix is now on trunk, and will be available in the 2.0.0 release targeted for February, I strongly believe it's time to make the functionality available in the next 1.9.3 patch release (and RubyInstaller release) in order to get more widespread usage.	
By backporting this functionality to the next 1.9.3 patch release and getting more usage (and potentially, further refinements) we minimize risks and help enhance the smooth adoption of 2.0.0.	

Associated revisions

Revision 2c400078 - 10/25/2012 08:16 AM - usa (Usaku NAKAMURA)

merge revision(s) 34849,34853,34854,34855,34859,34862,35384,35385,36811,36812,36850,36907,36908: [Backport #7174]

```
* Makefile.in (PLATFORM_DIR): add a variable for `win32` directory.
```

```
* Makefile.in (clean-platform): add new target.  
It cleans `win32` directory.
```

```
* common.mk (clean): add a dependency for `win32` directory.
```

```
* common.mk (distclean): ditto.
```

```
* common.mk (distclean-platform): add new target.  
It cleans `win32` directory.
```

```
* common.mk ($(PLATFORM_D)): add new target to make `win32` directory.
```

```
* common.mk (win32/win32.$(OBJEXT)): move win32.o into `win32`  
directory.
```

```
* common.mk (win32/file.$(OBJEXT)): add new target for win32/file.c.
```

```
* configure.in: move win32.o into `win32` directory and add
```

win32/file.o to MISSING.

- * file.c (file_load_ok, rb_file_load_ok): replace static file_load_ok() with public rb_file_load_ok(). It's to link Windows implementation in win32/file.c.
- * file.c (rb_find_file_ext_safe): ditto.
- * file.c (rb_find_file_safe): ditto.
- * win32/file.c (rb_file_load_ok): new file. Add Windows specific optimized implementation of rb_file_load_ok(). We created a separated file to avoid too many #ifdef macro which is unreadable.
- * win32/Makefile.sub (PLATFORM_DIR): add a variable for `win32` directory.
- * win32/Makefile.sub (MISSING): move win32.obj into `win32` directory and add win32/file.obj to MISSING.
- * win32/Makefile.sub (MAKEDIRS): replace MINIRUBY with BASERUBY. It's because miniruby doesn't exist when making `win32` directory.
- * win32/Makefile.sub (clean-platform): add new target to clean `win32` directory.
- * win32/Makefile.sub ({\${srcdir}}.c{}.obj): make it not match win32/file.c to build properly.
- * win32/Makefile.sub (win32/win32.\${OBJEXT}): move win32.obj into `win32` directory. Patch created with Luis Lavena. [ruby-core:42480] [Feature #5999]
- * win32/Makefile.sub (MAKEDIRS): use mkdir of cmd.exe instead of ruby. [Bug #6103] [ruby-core:43012]
- * win32/README.win32: added a notice about command extension of cmd.exe.
- * win32/makedirs.bat: new command to make intermediate directories, and not to report any errors if the directory already exists.
- * win32/Makefile.sub (MAKEDIRS): enable command extensions.
- * win32/file.c (INVALID_FILE_ATTRIBUTES): define for old SDK.
- * configure.in (mingw): add shlwapi to the list of dependency libs for Windows.
- * win32/Makefile.sub (EXTSOLIBS): ditto.
- * internal.h: declare internal functions rb_w32_init_file, rb_file_expand_path_internal and rb_file_expand_path_fast.
- * file.c (Init_File): invoke Windows initialization rb_w32_init_file
- * win32/file.c (rb_file_load_path_internal): new function. Windows-specific implementation that replaces file_expand_path. [Bug #6836][ruby-core:46996]
- * win32/file.c (rb_w32_init_file): new function. Initialize codepage cache for faster conversion encodings lookup.
- * file.c (file_expand_path): rename to rb_file_expand_path_internal. Conditionally exclude from Windows.
- * file.c (rb_file_expand_path_fast): new function. delegates to rb_file_expand_path_internal without performing a hit to the filesystem.
- * file.c (file_expand_path_1): use rb_file_expand_path_internal without path expansion (used by require).
- * file.c (rb_find_file_ext_safe): ditto.

- * file.c (rb_find_file_safe): ditto.
- * load.c (rb_get_expanded_load_path): use rb_file_expand_path_fast.
- * load.c (rb_feature_provided): ditto.
- * file.c (rb_file_expand_path): use rb_file_expand_path_internal with path expansion.
- * file.c (rb_file_absolute_path): ditto.
- * test/ruby/test_file_exhaustive.rb: new tests to exercise rb_file_expand_path_internal implementation and compliance with existing behaviors.
- * test/ruby/test_file_exhaustive.rb: fix test introduced in r36811 for posix environments where HOME is not defined. [ruby-core:47322]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_1_9_3@37321 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 37321 - 10/25/2012 08:16 AM - usa (Usaku NAKAMURA)

merge revision(s) 34849,34853,34854,34855,34859,34862,35384,35385,36811,36812,36850,36907,36908: [Backport #7174]

- * Makefile.in (PLATFORM_DIR): add a variable for `win32` directory.
- * Makefile.in (clean-platform): add new target. It cleans `win32` directory.
- * common.mk (clean): add a dependency for `win32` directory.
- * common.mk (distclean): ditto.
- * common.mk (distclean-platform): add new target. It cleans `win32` directory.
- * common.mk (\$(PLATFORM_D)): add new target to make `win32` directory.
- * common.mk (win32/win32.\$(OBJEXT)): move win32.o into `win32` directory.
- * common.mk (win32/file.\$(OBJEXT)): add new target for win32/file.c.
- * configure.in: move win32.o into `win32` directory and add win32/file.o to MISSING.
- * file.c (file_load_ok, rb_file_load_ok): replace static file_load_ok() with public rb_file_load_ok(). It's to link Windows implementation in win32/file.c.
- * file.c (rb_find_file_ext_safe): ditto.
- * file.c (rb_find_file_safe): ditto.
- * win32/file.c (rb_file_load_ok): new file. Add Windows specific optimized implementation of rb_file_load_ok(). We created a separated file to avoid too many #ifdef macro which is unreadable.
- * win32/Makefile.sub (PLATFORM_DIR): add a variable for `win32` directory.
- * win32/Makefile.sub (MISSING): move win32.obj into `win32` directory and add win32/file.obj to MISSING.
- * win32/Makefile.sub (MAKEDIRS): replace MINIRUBY with BASERUBY. It's because miniruby doesn't exist when making `win32` directory.
- * win32/Makefile.sub (clean-platform): add new target to clean `win32` directory.
- * win32/Makefile.sub ({\${srcdir}}.c{}.obj): make it not match win32/file.c to build properly.
- * win32/Makefile.sub (win32/win32.\$(OBJEXT)): move win32.obj into

```

`win32` directory.
Patch created with Luis Lavena.
[ruby-core:42480] [Feature #5999]

* win32/Makefile.sub (MAKEDIRS): use mkdir of cmd.exe instead of ruby.
[Bug #6103] [ruby-core:43012]

* win32/README.win32: added a notice about command extension of cmd.exe.

* win32/makedirs.bat: new command to make intermediate
directories, and not to report any errors if the directory
already exists.

* win32/Makefile.sub (MAKEDIRS): enable command extensions.

* win32/file.c (INVALID_FILE_ATTRIBUTES): define for old SDK.

* configure.in (mingw): add shlwapi to the list of dependency
libs for Windows.

* win32/Makefile.sub (EXTSOLIBS): ditto.

* internal.h: declare internal functions rb_w32_init_file,
rb_file_expand_path_internal and rb_file_expand_path_fast.

* file.c (Init_File): invoke Windows initialization rb_w32_init_file

* win32/file.c (rb_file_load_path_internal): new function.
Windows-specific implementation that replaces file_expand_path.
[Bug #6836][ruby-core:46996]

* win32/file.c (rb_w32_init_file): new function. Initialize codepage
cache for faster conversion encodings lookup.

* file.c (file_expand_path): rename to rb_file_expand_path_internal.
Conditionally exclude from Windows.

* file.c (rb_file_expand_path_fast): new function. delegates to
rb_file_expand_path_internal without performing a hit to the
filesystem.

* file.c (file_expand_path_1): use rb_file_expand_path_internal without
path expansion (used by require).

* file.c (rb_find_file_ext_safe): ditto.

* file.c (rb_find_file_safe): ditto.

* load.c (rb_get_expanded_load_path): use rb_file_expand_path_fast.

* load.c (rb_feature_provided): ditto.

* file.c (rb_file_expand_path): use rb_file_expand_path_internal with
path expansion.

* file.c (rb_file_absolute_path): ditto.

* test/ruby/test_file_exhaustive.rb: new tests to exercise
rb_file_expand_path_internal implementation and compliance with
existing behaviors.

* test/ruby/test_file_exhaustive.rb: fix test introduced in r36811 for
posix environments where HOME is not defined. [ruby-core:47322]

```

History

#1 - 10/19/2012 01:27 PM - usa (Usaku NAKAMURA)

- Status changed from Open to Assigned
- Assignee set to h.shirosaki (Hiroshi Shirosaki)

Shirosaki-san, can you make a patch for 1.9.3?
The base of r36811 is too far from 1.9.3.

#2 - 10/19/2012 10:53 PM - jonforums (Jon Forums)

Redmine is misbehaving.

Hiroshi...if it makes things simpler, please feel free to do a manual sync from a local svn repo to the tcs github repo so it's easy to rebase your current win-file-trunk/ruby_1_9_3 on top of the latest ruby_1_9_3. Don't worry about the other tcs patch branches being misaligned. I'll clean them up this weekend.

I'm in meetings all day, but had planned to tweak one of these two svn -> git sync scripts to sync trunk and ruby_1_9_3 between local svn repos (Jenkins workspace) and a local tcs-ruby git repo and push to gh.

<https://gist.github.com/3907291>

<https://bitbucket.org/jcreenaune/svn-git-mirror/src>

Issue [#7174](#) has been updated by usa (Usaku NAKAMURA).

Status changed from Open to Assigned
Assignee set to h.shirosaki (Hiroshi Shirosaki)

Shirosaki-san, can you make a patch for 1.9.3?

The base of r36811 is too far from 1.9.3.

Backport [#7174](#): Advocating for backporting 36811
<https://bugs.ruby-lang.org/issues/7174#change-31041>

Author: jonforums (Jon Forums)
Status: Assigned
Priority: Normal
Assignee: h.shirosaki (Hiroshi Shirosaki)
Category:
Target version:

Please consider backporting 36811 and refinements for the following reasons:

1) While it's debatable whether this patch fixes a Windows platform bug in the 1.9.x series or whether it's a performance refactoring, it is a fact that this patch fixes long-standing 1.9.x usability problems important to many Ruby on Windows users.

<https://gist.github.com/3242245>

2) For many months Hiroshi has maintained a 1.9.3 backport of this functionality, with the latest version available at https://github.com/thecodeshop/ruby/tree/win-file-trunk/ruby_1_9_3. This functionality has been part of multiple tcs-ruby experimental releases over the last 6-9 months.

3) While the functionality has been tested on 1.9.3 via tcs-ruby by many interested devs over many months, tcs-ruby is not (and will never be) an alternative MRI distribution. tcs-ruby's sole purpose is to be collaboration repo and proving ground for those interested in contributing Windows and cross-platform improvements to MRI. tcs-ruby's goal is to refine the improvements and make them available for easy inclusion into MRI.

4) As a consequence of (3), Hiroshi's 1.9.3 backports have not had as much widespread testing as they would have if they were included in an official RubyInstaller release. This was a purposeful decision on our part. RubyInstaller releases are meant to track official ruby-core releases. They are meant to be stable, not experimental releases.

As this fix is now on trunk, and will be available in the 2.0.0 release targeted for February, I strongly believe it's time to make the functionality available in the next 1.9.3 patch release (and RubyInstaller release) in order to get more widespread usage.

By backporting this functionality to the next 1.9.3 patch release and getting more usage (and potentially, further refinements) we minimize risks and help enhance the smooth adoption of 2.0.0.

--

<http://bugs.ruby-lang.org/>

Jon

Fail fast. Fail often. Fail publicly. Learn. Adapt. Repeat.
<http://thecodeshop.github.com> | <http://jonforums.github.com/>
twitter: [jonforums \(Jon Forums\)](#)

#3 - 10/19/2012 11:59 PM - h.shirosaki (Hiroshi Shirosaki)

On Fri, Oct 19, 2012 at 1:27 PM, usa (Usaku NAKAMURA)
usa@garbagecollect.jp wrote:

Shirosaki-san, can you make a patch for 1.9.3?
The base of r36811 is too far from 1.9.3.

I've rebased on the top of [r37266](#). Recent revisions don't conflict.
I added two commits.

https://github.com/thecodeshop/ruby/commits/win-file-trunk/ruby_1_9_3
(I also updated ruby_1_9_3.)

Revisions:
r34849,r34853,r34854,r34855,r34859,r34862,r35384,r35385,r36811,r36812,r36850,r36907,r36908

Patch:
https://github.com/thecodeshop/ruby/compare/ruby_1_9_3...win-file-trunk/ruby_1_9_3.diff
<https://gist.github.com/3918535>

I confirmed make, make test and make test-all with the following
environments. It seems fine.
x86_64-darwin12.2.0, i386-mingw32, x64-mswin64

ruby 1.9.3p297 [i386-mingw32] on Win7
has two failures. Other fixes will be required for those.

1) Failure:
test_getpwuid(TestEtc) [c:/Users/hiroshi/work/ruby/test/etc/test_etc.rb:42]:
Expected [] to include nil.

2) Failure:
test_generate_bin_bindir_with_user_install_warning(TestGemInstaller)
[c:/Users/hiroshi/work/ruby/test/rubygems/test_gem_
installer.rb:247]:
--- expected
+++ actual
@@ -1 +1,3 @@
_""
+"WARNING: You don't have C:\Windows in your PATH,
+\t gem executables will not run.
+"

10356 tests, 1909402 assertions, 2 failures, 0 errors, 94 skips

r36636 will fix test_getpwuid failure.
r37048 will fix test_generate_bin_bindir_with_user_install_warning failure.
r37032 will fix parallel test-all failure.

Thank you.
--
Hiroshi Shirosaki

#4 - 10/25/2012 05:16 PM - usa (Usaku NAKAMURA)

- Status changed from Assigned to Closed
- % Done changed from 0 to 100

This issue was solved with changeset [r37321](#).
Jon, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

merge revision(s) 34849,34853,34854,34855,34859,34862,35384,35385,36811,36812,36850,36907,36908: [Backport [#7174](#)]

```
* Makefile.in (PLATFORM_DIR): add a variable for `win32` directory.  
  
* Makefile.in (clean-platform): add new target.  
  It cleans `win32` directory.  
  
* common.mk (clean): add a dependency for `win32` directory.
```

```

* common.mk (distclean): ditto.

* common.mk (distclean-platform): add new target.
  It cleans `win32` directory.

* common.mk ($(PLATFORM_D)): add new target to make `win32` directory.

* common.mk (win32/win32.$(OBJEXT)): move win32.o into `win32`
  directory.

* common.mk (win32/file.$(OBJEXT)): add new target for win32/file.c.

* configure.in: move win32.o into `win32` directory and add
  win32/file.o to MISSING.

* file.c (file_load_ok, rb_file_load_ok): replace static
  file_load_ok() with public rb_file_load_ok().
  It's to link Windows implementation in win32/file.c.

* file.c (rb_find_file_ext_safe): ditto.

* file.c (rb_find_file_safe): ditto.

* win32/file.c (rb_file_load_ok): new file. Add Windows specific
  optimized implementation of rb_file_load_ok(). We created a
  separated file to avoid too many #ifdef macro which is unreadable.

* win32/Makefile.sub (PLATFORM_DIR): add a variable for `win32`
  directory.

* win32/Makefile.sub (MISSING): move win32.obj into `win32`
  directory and add win32/file.obj to MISSING.

* win32/Makefile.sub (MAKEDIRS): replace MINIRUBY with BASERUBY.
  It's because miniruby doesn't exist when making `win32` directory.

* win32/Makefile.sub (clean-platform): add new target to clean `win32`
  directory.

* win32/Makefile.sub ({${srcdir}}.c{}.obj): make it not match
  win32/file.c to build properly.

* win32/Makefile.sub (win32/win32.$(OBJEXT)): move win32.obj into
  `win32` directory.
  Patch created with Luis Lavena.
  [ruby-core:42480] [Feature #5999]

* win32/Makefile.sub (MAKEDIRS): use mkdir of cmd.exe instead of ruby.
  [Bug #6103] [ruby-core:43012]

* win32/README.win32: added a notice about command extension of cmd.exe.

* win32/makedirs.bat: new command to make intermediate
  directories, and not to report any errors if the directory
  already exists.

* win32/Makefile.sub (MAKEDIRS): enable command extensions.

* win32/file.c (INVALID_FILE_ATTRIBUTES): define for old SDK.

* configure.in (mingw): add shlwapi to the list of dependency
  libs for Windows.

* win32/Makefile.sub (EXTSOLIBS): ditto.

* internal.h: declare internal functions rb_w32_init_file,
  rb_file_expand_path_internal and rb_file_expand_path_fast.

* file.c (Init_File): invoke Windows initialization rb_w32_init_file

* win32/file.c (rb_file_load_path_internal): new function.
  Windows-specific implementation that replaces file_expand_path.
  [Bug #6836] [ruby-core:46996]

```

* win32/file.c (rb_w32_init_file): new function. Initialize codepage cache for faster conversion encodings lookup.

* file.c (file_expand_path): rename to rb_file_expand_path_internal. Conditionally exclude from Windows.

* file.c (rb_file_expand_path_fast): new function. delegates to rb_file_expand_path_internal without performing a hit to the filesystem.

* file.c (file_expand_path_1): use rb_file_expand_path_internal without path expansion (used by require).

* file.c (rb_find_file_ext_safe): ditto.

* file.c (rb_find_file_safe): ditto.

* load.c (rb_get_expanded_load_path): use rb_file_expand_path_fast.

* load.c (rb_feature_provided): ditto.

* file.c (rb_file_expand_path): use rb_file_expand_path_internal with path expansion.

* file.c (rb_file_absolute_path): ditto.

* test/ruby/test_file_exhaustive.rb: new tests to exercise rb_file_expand_path_internal implementation and compliance with existing behaviors.

* test/ruby/test_file_exhaustive.rb: fix test introduced in r36811 for posix environments where HOME is not defined. [ruby-core:47322]

#5 - 10/25/2012 10:38 PM - h.shirosaki (Hiroshi Shirosaki)

- Status changed from Closed to Assigned

- Assignee changed from h.shirosaki (Hiroshi Shirosaki) to usa (Usaku NAKAMURA)

Thank you very much for backport!

But I noticed the following failure occurs.

<http://u64.rubyci.org/~chkbuild/ruby-1.9.3/log/20121025T112858Z.log.html.gz>

28) Error:

test_expand_path_encoding(TestFileExhaustive):

Encoding::InvalidByteSequenceError: "\xA4" on US-ASCII

/home/chkbuild/build/ruby-1.9.3/20121025T112858Z/ruby/test/ruby/test_file_exhaustive.rb:433:in encode'

/home/chkbuild/build/ruby-1.9.3/20121025T112858Z/ruby/test/ruby/test_file_exhaustive.rb:433:intest_expand_path_encoding'

Since ruby_1_9_3 doesn't have encoding support of File.expand_path, I think skipping the test might be better.

Encoding fix seems introduced at [#5919](#) (r34372), but r34372 was not merged according to the list.

<http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-dev/45903>

In my proposed backport patch skipping the test if non-windows platform.

https://github.com/thecodeshop/ruby/blob/win-file-trunk/ruby_1_9_3/test/ruby/test_file_exhaustive.rb#L439

#6 - 10/26/2012 10:53 AM - usa (Usaku NAKAMURA)

Hello,

In message "[ruby-core:48282] [Backport93 - Backport [#7174](#)][Assigned] Advocating for backporting 36811" on Oct.25,2012 22:38:02, h.shirosaki@gmail.com wrote:

But I noticed the following failure occurs.

<http://u64.rubyci.org/~chkbuild/ruby-1.9.3/log/20121025T112858Z.log.html.gz>

Thanks.

Since ruby_1_9_3 doesn't have encoding support of File.expand_path, I think skipping the test might be better. Encoding fix seems introduced at [#5919](#) (r34372), but r34372 was not merged according to the list.

<http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-dev/45903>

In my proposed backport patch skipping the test if non-windows platform.

https://github.com/thecodeshop/ruby/blob/win-file-trunk/ruby_1_9_3/test/ruby/test_file_exhaustive.rb#L439

I'm planning to backport r34372.

It was not rejected, but was only postponed because it's difficult.

So, please wait a little.

Regards,

--

U.Nakamura usa@garbagecollect.jp

#7 - 11/08/2012 02:07 PM - usa (Usaku NAKAMURA)

- *Status changed from Assigned to Closed*

already backported.