Ruby master - Bug #5637

warnings of shellescape

11/15/2011 04:29 PM - znz (Kazuhiro NISHIYAMA)

Status: Closed
Priority: Normal
Assignee: knu (Akinori MUSHA)
Target version: 2.0.0
ruby -v: -

Description

Associated revisions

Revision d367b1b9 - 01/01/2012 03:26 AM - knu (Akinori MUSHA)

- lib/shellwords.rb (Shellwords#shellescape): Drop the /n flag that only causes warnings with no real effect. [Bug #5637]
git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@34166 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 34166 - 01/01/2012 03:26 AM - knu (Akinori MUSHA)

- lib/shellwords.rb (Shellwords#shellescape): Drop the /n flag that only causes warnings with no real effect. [Bug #5637]

Revision 34166 - 01/01/2012 03:26 AM - knu (Akinori MUSHA)

- lib/shellwords.rb (Shellwords#shellescape): Drop the /n flag that only causes warnings with no real effect. [Bug #5637]

Revision 34166 - 01/01/2012 03:26 AM - knu (Akinori MUSHA)

- lib/shellwords.rb (Shellwords#shellescape): Drop the /n flag that only causes warnings with no real effect. [Bug #5637]

Revision 34166 - 01/01/2012 03:26 AM - knu (Akinori MUSHA)

- lib/shellwords.rb (Shellwords#shellescape): Drop the /n flag that only causes warnings with no real effect. [Bug #5637]

Revision b760f18a - 02/12/2012 07:38 AM - knu (Akinori MUSHA)

merge revision(s) 34166:

- * lib/shellwords.rb (Shellwords#shellescape): Drop the /n flag that only causes warnings with no real effect. [Bug #5637]
git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_1_9_3@34570 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 11/15/2011 04:53 PM - znz (Kazuhiro NISHIYAMA)
- ruby -v changed from ruby 2.0.0dev (2011-11-15 trunk 33753) [x86_64-linux] to -
Shellwords.shellescape で警告が出ます。

% ./ruby -v -r shellwords -e 'p Shellwords.shellescape("\\u3042")
ruby 2.0.0dev (2011-11-15 trunk 33753) [x86_64-linux]
/home/chkbuild/tmp/build/ruby-trunk/20111114T222552Z/lib/ruby/1.9.1/shellwords.rb:86: warning: regexp match /.../n against to UTF-8 string
/home/chkbuild/tmp/build/ruby-trunk/20111114T222552Z/lib/ruby/1.9.1/shellwords.rb:86: warning: regexp match /.../n against to UTF-8 string
"あ"

エスケープ結果も変だと思います。

エスケープ結果を 1.8.7 にあわせるのなら以下のパッチで
どうでしょうか。

diff --git a/lib/shellwords.rb b/lib/shellwords.rb
index 5d6ba75..78331a7 100644
--- a/lib/shellwords.rb
+++ b/lib/shellwords.rb
@@ -79,11 +79,11 @@ module Shellwords
          # A LF cannot be escaped with a backslash because a backlash + LF
          # combo is regarded as line continuation and simply ignored.
          diff --git a/test/test_shellwords.rb b/test/test_shellwords.rb
@@ -36,4 +36,8 @@ class TestShellwords < Test::Unit::TestCase
               shellwords(bad_cmd)
               end
           end
+         def test_shellescape_utf8_string
+           assert_equal "\343\201\202", shellescape("\u3042")
+         end
+         end

#2 - 11/16/2011 08:24 PM - knu (Akinori MUSHA)
- Assignee set to knu (Akinori MUSHA)

#3 - 12/22/2011 07:51 PM - knu (Akinori MUSHA)

#4 - 01/01/2012 12:26 PM - knu (Akinori MUSHA)
This issue was solved with changeset r34166.
Kazuhiro, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- lib/shellwords.rb (Shellwords#shellescape): Drop the /n flag that only causes warnings with no real effect. [Bug #5637]

#5 - 08/07/2012 02:48 AM - dariocravero (Darío Cravero)
Hi,

Thanks for this patch!... :)

One question though, from comment #3 it's not clear if it's safe to use it in 1.9.3. This is what Google Translator gave me:

"1.9: this behavior was all the way to 1.9.3 now. Turn off warning but does not change as a bug (missing fix of / / n), because the behavior leads to incompatibility."

However, I've applied it and, as expected, I don't see the warning anymore. Still, can you just confirm there're no side effects to this on 1.9.3?

Thanks a million!

#6 - 08/08/2012 08:25 PM - knu (Akinori MUSHA)
As I documented, it's all up to how you use the resulted string.

If you are going to pass it to a shell that lacks support for the encoding of the string, then you should probably encode the original string in ASCII-8BIT before shell-escaping with shellescape() to get a byte-by-byte escape to make sure the shell won't find a metacharacter inside a multibyte character.

UTF-8 multibyte characters do not contain any ASCII character by design anyway, so most people in the everything-is-UTF-8 world don't even have to care about this.

But, for example, when you have to run a program passing a Shift_JIS string via a shell under a non-Shift_JIS locale, you'd probably have to compose the command line in the ASCII-8BIT encoding so that all shell metacharacters that may appear in Shift_JIS multibyte characters are properly escaped.