

Ruby master - Feature #4477

Kernel:exec and backtick (`) don't work for certain system commands

03/07/2011 05:55 PM - JWuttke (Joachim Wuttke)

Status:	Closed	
Priority:	Normal	
Assignee:	akr (Akira Tanaka)	
Target version:		
Description		
<p>From documentation and common sense, I would expect that Kernel:exec and backtick (`) work for <u>all</u> kind of system commands. This is currently not the case.</p> <pre>~\$ cat test1 echo success ~\$ cat test2 #!/bin/sh echo success ~\$ chmod a+x test? ~\$ ruby1.8 -e 'p test1' -e:1: command not found: test1 "" ~\$ ruby1.9 -e 'p test1' -e:1:in `': Exec format error - test1 (Errno::ENOEXEC) from -e:1:in ` ~\$ ruby1.8 -e 'p test2' "success\n" ~\$ ruby1.9 -e 'p test2' "success\n"</pre> <p>This problem has been reported before (#3856: strange Kernel#exec behavior with bash's source command). As a workaround, it has been suggested to append a semikolon to the system command:</p> <pre>~\$ ruby1.8 -e 'p test1;' "success\n" ~\$ ruby1.9 -e 'p test1;' "success\n"</pre> <p>The report #3856 has been closed with the decision "not a bug".</p> <p>This I cannot accept. For many years, I got used to run system commands through <cmd>. When it failed, I spent painfully long time to search for a bug in my Ruby code and in the system command before I realized the problem was due to an unexpected restriction of Ruby's exec/backtick.</p>		
Related issues:		
Related to Ruby master - Feature #2715: Optimization to avoid spawning shell ...	Feedback	
Related to Ruby master - Feature #4269: command with && should execute as shell	Closed	01/12/2011

History

#1 - 03/07/2011 06:53 PM - sorah (Sorah Fukumori)

- Tracker changed from Bug to Feature

=begin

=end

#2 - 03/07/2011 07:40 PM - JWuttke (Joachim Wuttke)

=begin

Sorry, Shota, I do not understand why you recategorized this issue as a Feature.

The current behaviour of <cmd> contradicts what the documentation says, and it offends common sense. In my understanding, this is clearly a Bug.

=end

#3 - 03/07/2011 09:21 PM - sorah (Sorah Fukumori)

=begin

I don't think this is Bug because that (Shebang less executing) is you shell's feature. not system's feature.

So this ticket should be Feature request.

=end

#4 - 03/07/2011 10:23 PM - kosaki (Motohiro KOSAKI)

=begin

2011/3/7 Shota Fukumori sorah@tubusu.net:

Issue [#4477](#) has been updated by Shota Fukumori.

I don't think this is Bug because that (Shebang less executing) is you shell's feature. not system's feature.

No. it is POSIX spec. But, FWIW, This behavior have been changed intentionally. So, I doubt the issue is a documentation fault.

Currently `don't use shell if command line string don't have following shell meta characters.

```
*?{}[]<>()~&|\$;"'\n
```

But, Which documentation describe it?

=end

#5 - 03/18/2012 07:06 PM - akr (Akira Tanaka)

- Description updated

#6 - 03/25/2012 03:05 PM - mame (Yusuke Endoh)

- Status changed from Open to Assigned

- Assignee set to kosaki (Motohiro KOSAKI)

#7 - 03/25/2012 03:07 PM - mame (Yusuke Endoh)

- Assignee changed from kosaki (Motohiro KOSAKI) to akr (Akira Tanaka)

#8 - 03/25/2012 03:07 PM - mrkn (Kenta Murata)

- Description updated

#9 - 06/04/2012 08:53 PM - akr (Akira Tanaka)

[#3856](#) is at RubyForge tracker:

http://rubyforge.org/tracker/?func=detail&atid=1698&aid=3856&group_id=426

#10 - 06/04/2012 08:57 PM - akr (Akira Tanaka)

- Status changed from Assigned to Closed

The feature of shell, invoke /bin/sh after ENOEXEC, is implemented at r33009 of Ruby.

So this example works now as follows.

```
% cat test1
echo success
% chmod a+x test1
% ./ruby -ve 'p `./test1`'
ruby 2.0.0dev (2012-06-04 trunk 35911) [x86_64-linux]
"success\n"
```