

## Ruby master - Bug #8133

### Regexp macro %r{} is loosing backslash on "\"

03/21/2013 12:22 AM - jstribny (Josef Stribny)

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>		
<b>Target version:</b>	2.1.0	
<b>ruby -v:</b>	2.0.0	<b>Backport:</b>
<b>Description</b>		
Hello,		
Ruby 2.0.0 is loosing the backslash when used on "\"" string (exact match).		
How to reproduce in irb:		
2.0.0-p0 :003 > %r{}		
=> /{/		
2.0.0-p0 :004 > %r{}		
=> /{/		
Compared to:		
1.9.3p392 :001 > %r{}		
=> /{/		
1.9.3p392 :002 > %r{}		
=> /{/		
This shouldn't probably happen.		
<b>Related issues:</b>		
Related to Ruby master - Bug #7610: Incompatible with percent literal regexp ...	<b>Closed</b>	<b>12/23/2012</b>
Has duplicate Backport200 - Backport #8135: Backport escape all closing paren...	<b>Closed</b>	<b>03/21/2013</b>

#### Associated revisions

##### Revision ebd86634 - 03/21/2013 07:48 AM - nobu (Nobuyoshi Nakada)

parse.y: escape all closing parens

- parse.y (simple\_re\_meta): escape all closing characters, not only round parenthesis. [ruby-core:53578] [Bug #8133]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@39858 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 39858 - 03/21/2013 07:48 AM - nobu (Nobuyoshi Nakada)

parse.y: escape all closing parens

- parse.y (simple\_re\_meta): escape all closing characters, not only round parenthesis. [ruby-core:53578] [Bug #8133]

##### Revision 39858 - 03/21/2013 07:48 AM - nobu (Nobuyoshi Nakada)

parse.y: escape all closing parens

- parse.y (simple\_re\_meta): escape all closing characters, not only round parenthesis. [ruby-core:53578] [Bug #8133]

##### Revision 39858 - 03/21/2013 07:48 AM - nobu (Nobuyoshi Nakada)

parse.y: escape all closing parens

- parse.y (simple\_re\_meta): escape all closing characters, not only round parenthesis. [ruby-core:53578] [Bug #8133]

##### Revision 39858 - 03/21/2013 07:48 AM - nobu (Nobuyoshi Nakada)

parse.y: escape all closing parens

- `parse.y (simple_re_meta)`: escape all closing characters, not only round parenthesis. [ruby-core:53578] [Bug #8133]

#### Revision 39858 - 03/21/2013 07:48 AM - nobu (Nobuyoshi Nakada)

`parse.y`: escape all closing parens

- `parse.y (simple_re_meta)`: escape all closing characters, not only round parenthesis. [ruby-core:53578] [Bug #8133]

#### Revision 39858 - 03/21/2013 07:48 AM - nobu (Nobuyoshi Nakada)

`parse.y`: escape all closing parens

- `parse.y (simple_re_meta)`: escape all closing characters, not only round parenthesis. [ruby-core:53578] [Bug #8133]

#### Revision 6c0822f0 - 04/06/2013 06:07 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 39858: [Backport #8135]

```
* parse.y (simple_re_meta): escape all closing characters, not only
  round parenthesis. [ruby-core:53578] [Bug #8133]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby\_2\_0\_0@40169 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

## History

---

### #1 - 03/21/2013 11:26 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Rejected

There is nothing different, as '}' is not a regexp meta character.

### #2 - 03/21/2013 04:13 PM - vo.x (Vit Ondruch)

- Status changed from Rejected to Open

It doesn't look to be issue of regexp but issue of parser:

```
C:\Projects>irb
irb(main):001:0> a = %r{}}
=> /}/
irb(main):002:0> b = %r{}}
=> /}/
irb(main):003:0> a == b
=> false
```

a and b are apparently not the same. They used to be the same in Ruby 1.9.3.

### #3 - 03/21/2013 04:48 PM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r39858.

Josef, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

---

`parse.y`: escape all closing parens

- `parse.y (simple_re_meta)`: escape all closing characters, not only round parenthesis. [ruby-core:53578] [Bug #8133]

### #4 - 03/21/2013 05:46 PM - vo.x (Vit Ondruch)

nobu (Nobuyoshi Nakada) wrote:

This issue was solved with changeset r39858.

Thank you.

### #5 - 03/21/2013 11:41 PM - mame (Yusuke Endoh)

Just for information: r39858 introduces the behavior change.

Whether `%r{n{3}}` matches with:

	"nnn"	"n{3}"
1.9.3p392	YES	NO
2.0.0p0	NO	YES
trunk	YES	NO

--

Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)

**#6 - 03/22/2013 01:40 AM - nagachika (Tomoyuki Chikanaga)**

Hmm, is it a bug introduced in 2.0.0?  
I personally feel the behavior of 2.0.0 is natural.

**#7 - 03/22/2013 04:26 PM - naruse (Yui NARUSE)**

nagachika (Tomoyuki Chikanaga) wrote:

    Hmm, is it a bug introduced in 2.0.0?  
    I personally feel the behavior of 2.0.0 is natural.

} in this case doesn't escape a regexp meta character, it escapes closing terminator of `%r{...}` literal.

**#8 - 03/25/2013 02:56 AM - nagachika (Tomoyuki Chikanaga)**

mame (Yusuke Endoh) wrote:

    Just for information: r39858 introduces the behavior change.  
    Whether `%r{n{3}}` matches with:

	"nnn"	"n{3}"
1.9.3p392	YES	NO
2.0.0p0	NO	YES
trunk	YES	NO

On my environment the result was exactly opposite.

```
$ cat regexp_test.rb
re = %r{n{3}}
p re
p [re =~ "nnn", re =~ "n{3}"]
```

```
ruby 1.9.3dev (2011-09-24 revision 33322) [x86_64-darwin10.8.0]
/n{3}/
[nil, 0]
```

```
ruby 2.0.0p96 (2013-03-25 revision 39918) [x86_64-darwin10.8.0]
/n{3}/
[0, nil]
```

```
ruby 2.1.0dev (2013-03-24 trunk 39908) [x86_64-darwin10.8.0]
/n{3}/
[nil, 0]
```

**#9 - 03/25/2013 02:59 AM - nagachika (Tomoyuki Chikanaga)**

naruse (Yui NARUSE) wrote:

} in this case doesn't escape a regexp meta character, it escapes closing terminator of `%r{...}` literal.

I agree. It's reasonable.  
Thanks.