Ruby master - Bug #7380
benchmark/driver.rb: --executables separator
11/17/2012 08:15 PM - Eregon (Benoit Daloze)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>ko1 (Koichi Sasada)</td>
</tr>
<tr>
<td>Target version:</td>
<td>2.0.0</td>
</tr>
</tbody>
</table>

Description
Hello,

The separator used between ruby executables for the benchmark/driver.rb script is ";".
This requires the argument to be quoted (at least in bash/zsh) and produce an unintuitive result if not: only the first executable is taken by the script, the second executable launches the ruby interpreter with no arguments, thus waiting on stdin, which feels like the benchmark script hangs.

I proposed to use ";", the standard path separator instead.
For compatibility, ";" could split too, but I wish to document ";" in the help.

What do you think? May I go ahead?

Associated revisions
Revision 2e87867a - 11/26/2012 09:19 AM - ko1 (Koichi Sasada)
- benchmark/driver.rb: accept multiple -e'. You don't need to use; separation character. [ruby-core:50139] [ruby-trunk - Bug #7380]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@37854 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 37854 - 11/26/2012 09:19 AM - ko1 (Koichi Sasada)
- benchmark/driver.rb: accept multiple -e'. You don't need to use; separation character. [ruby-core:50139] [ruby-trunk - Bug #7380]

Revision 37854 - 11/26/2012 09:19 AM - ko1 (Koichi Sasada)
- benchmark/driver.rb: accept multiple -e'. You don't need to use; separation character. [ruby-core:50139] [ruby-trunk - Bug #7380]

Revision 37854 - 11/26/2012 09:19 AM - ko1 (Koichi Sasada)
- benchmark/driver.rb: accept multiple -e'. You don't need to use; separation character. [ruby-core:50139] [ruby-trunk - Bug #7380]

Revision 37854 - 11/26/2012 09:19 AM - ko1 (Koichi Sasada)
- benchmark/driver.rb: accept multiple -e'. You don't need to use; separation character. [ruby-core:50139] [ruby-trunk - Bug #7380]

Revision 37854 - 11/26/2012 09:19 AM - ko1 (Koichi Sasada)
- benchmark/driver.rb: accept multiple -e'. You don't need to use; separation character. [ruby-core:50139] [ruby-trunk - Bug #7380]

Revision 37854 - 11/26/2012 09:19 AM - ko1 (Koichi Sasada)
- benchmark/driver.rb: accept multiple -e'. You don't need to use; separation character. [ruby-core:50139] [ruby-trunk - Bug #7380]

Revision 37854 - 11/26/2012 09:19 AM - ko1 (Koichi Sasada)
- benchmark/driver.rb: accept multiple -e'. You don't need to use; separation character. [ruby-core:50139] [ruby-trunk - Bug #7380]

Revision 37854 - 11/26/2012 09:19 AM - ko1 (Koichi Sasada)
- benchmark/driver.rb: accept multiple -e'. You don't need to use; separation character. [ruby-core:50139] [ruby-trunk - Bug #7380]

Revision 37854 - 11/26/2012 09:19 AM - ko1 (Koichi Sasada)
- benchmark/driver.rb: accept multiple -e'. You don't need to use; separation character. [ruby-core:50139] [ruby-trunk - Bug #7380]

Revision 37854 - 11/26/2012 09:19 AM - ko1 (Koichi Sasada)
- benchmark/driver.rb: accept multiple -e'. You don't need to use; separation character. [ruby-core:50139] [ruby-trunk - Bug #7380]

History
#1 - 11/24/2012 06:40 PM - mame (Yusuke Endoh)
- Status changed from Open to Assigned
ko1, what do you think?

--
Yusuke Endoh mame@tsg.ne.jp

#2 - 11/26/2012 09:52 AM - ko1 (Koichi Sasada)
I always use -e with quote strings. does it bother you?

By the way, -e accept `:`
like -e "foo::path1; bar::path2"
as a label of path.

#3 - 11/26/2012 06:19 PM - ko1 (Koichi Sasada)
- Status changed from Assigned to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r37854.
Benoit, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

• benchmark/driver.rb: accept multiple -e'. You don't need to use;` separation character. [ruby-core:50139] [ruby-trunk - Bug #7380]

#4 - 11/26/2012 10:12 PM - Eregon (Benoit Daloze)
ko1 (Koichi Sasada) wrote:

I always use -e with quote strings. does it bother you?

By the way, -e accept `:`
like -e "foo::path1; bar::path2"
as a label of path.

Oh, right, I did not think to the name::path usage, it would be weird to have both :: and :

I like the solution to pass multiple -e options, thank you!