Description

When I call DRb.start_service('druby://localhost:1234') and port 1234 is already in use, I silently end up with a broken DRb service. start_service should detect that the port is already in use and raise an exception.

I've attached a small test case. Starting it once works fine. But keeping it running and starting it again from another shell should fail. It does not.

History

#1 - 04/22/2010 11:51 PM - mame (Yusuke Endoh)

=begin
Umm, I cannot reproduce.
This issue may depend on network configuration...

# terminal 1
$ ./ruby drb_port_busy_bug.rb
druby://localhost:1234
Server is alive

# terminal 2
$ ./ruby drb_port_busy_bug.rb
/home/mame/work/ruby-trunk-local/lib/ruby/1.9.1/drb/drb.rb:861:in initialize': Address family not supported by protocol - socket(2)
(Errno::EAFNOSUPPORT)
from /home/mame/work/ruby-trunk-local/lib/ruby/1.9.1/drb/drb.rb:861:inopen'
from /home/mame/work/ruby-trunk-local/lib/ruby/1.9.1/drb/drb.rb:861:in open_server'
from /home/mame/work/ruby-trunk-local/lib/ruby/1.9.1/drb/drb.rb:759:inblock in open_server'
from /home/mame/work/ruby-trunk-local/lib/ruby/1.9.1/drb/drb.rb:757:in each'
from /home/mame/work/ruby-trunk-local/lib/ruby/1.9.1/drb/drb.rb:757:inopen_server'
from /home/mame/work/ruby-trunk-local/lib/ruby/1.9.1/drb/drb.rb:1342:in initialize'
from /home/mame/work/ruby-trunk-local/lib/ruby/1.9.1/drb/drb.rb:1630:innew'
from /home/mame/work/ruby-trunk-local/lib/ruby/1.9.1/drb/drb.rb:1630:start_service'
from drb_port_busy_bug.rb:3:in'

$ ./ruby -v
ruby 1.9.2dev (2010-04-22 trunk 27445) [i686-linux]

--
Yusuke Endoh mame@tsg.ne.jp
=end

#2 - 04/23/2010 12:06 AM - orem (Chris Schlaeger)

=begin
I did some more tests and the problem is most likely not DRb specific.
I can reproduce this with TCPSocket as well. The system was an openSUSE 11.1 and 11.2. Both have IPv6 enabled. The problem does not occur when I use 127.0.0.1 instead of localhost. Does that help?
Should I do another test?

Chris
=end
Hi,

I did some more tests and the problem is most likely not DRb specific. I can reproduce this with TCPSocket as well. The system was an openSUSE 11.1 and 11.2. Both have IPv6 enabled. The problem does not occur when I use 127.0.0.1 instead of localhost. Does that help? Should I do another test?

This may be side effect of ipv6 support of ext/socket. What's printed by:

```ruby
require "socket"
p TCPServer.new("localhost", 1234).addr
p TCPServer.new("localhost", 1234).addr
```

If the result is, for example:

```ruby
["AF_INET6", 1234, ":1", ":1"]
["AF_INET", 1234, "127.0.0.1", "127.0.0.1"]
```

, this is intended behavior.

The first drb server is bound to ::1:1234, and the second is 127.0.0.1:1234. You will be able to access both servers to druby://[::1]:1234 and druby://127.0.0.1:1234, respectively.

--
Yusuke Endoh mame@tsg.ne.jp

# ruby19 socket-ipv6.rb
["AF_INET6", 1234, "localhost", ":1"]
["AF_INET", 1234, "127.0.0.1", "127.0.0.1"]

# host localhost
localhost has address 127.0.0.1

Hope that helps. I can run more tests if needed.
Chris
Hi,

This is intended behavior.

You can actually run two drb servers at the time on your environment. One can be accessed via druby://[::1]:1234, and the other can be druby://127.0.0.1:1234. You will see error when attempting to invoke the third drb server. There is problem nowhere.

If you dislike this behavior, use druby://127.0.0.1:1234 as server URL, instead of druby://localhost:1234. Or, disabling ipv6 on your SuSE configuration will also work, though I don't encourage you to do it.

And, thank you for your quick response!

--
Yusuke Endoh mame@tsg.ne.jp

Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>drb_port_busy_bug.rb</td>
<td>131 B</td>
<td>03/30/2010</td>
<td>orem (Chris Schlaeger)</td>
</tr>
</tbody>
</table>