Ruby master - Bug #5618

Exceptions cause DRb connection to be closed

11/11/2011 10:49 PM - larsch (Lars Christensen)

Status:	Closed		
Priority:	Normal		
Assignee:	seki (Masatoshi Seki)		
Target version:			
ruby -v:	ruby 1.9.3p0 (2011-10-30) [i386-mingw32]	**Backport:**	

Description

If an exception is thrown in a DRb server of a class that is not defined in the client process, the DRb connection is closed abrubtly. In Ruby 1.8, the exception was translated to an instance generic exception class (DRb::DRbRemoteError).

Steps to reproduce:

1. Start DRb client+server
2. Throw exception of custom class in server (class not defined in client script).

Observed behaviour:

1. Client raises DRb::DRbConnError because connection is closed - used to be generic error.

Expected behaviour:

1. Client raises DRb::DRbRemoteError, and preserves full stack trace from server.

Attached is server/client script showing the problem.

Here is the stack trace of the client:

```
C:/lang/Ruby193/lib/ruby/1.9.1/drb/drb.rb:570:in load': connection closed (DRb::DRbConnError)
from C:/lang/Ruby193/lib/ruby/1.9.1/drb/drb.rb:632:inrecv_reply'
from C:/lang/Ruby193/lib/ruby/1.9.1/drb/drb.rb:918:in recv_reply'
from C:/lang/Ruby193/lib/ruby/1.9.1/drb/drb.rb:1197:insend_message'
from C:/lang/Ruby193/lib/ruby/1.9.1/drb/drb.rb:1088:in block (2 levels) in method_missing'
from C:/lang/Ruby193/lib/ruby/1.9.1/drb/drb.rb:1172:inopen'
from C:/lang/Ruby193/lib/ruby/1.9.1/drb/drb.rb:1087:in block in method_missing'
from C:/lang/Ruby193/lib/ruby/1.9.1/drb/drb.rb:1105:inwith_friend'
from C:/lang/Ruby193/lib/ruby/1.9.1/drb/drb.rb:1086:in method_missing'
from drbclient.rb:3:in'
```

Associated revisions

Revision 50b8033d - 10/16/2019 07:51 PM - jeremyevans (Jeremy Evans)

Handle subclasses of Exception in drb

This makes it so that client instances that call a method on a server-side object that raise a subclass of Exception are handled similarly to those that raise a subclass of StandardError.

Only reraise exceptions that we are fairly sure we don't want to rescue.

Fixes [Bug #5618]

History

#1 - 03/11/2012 04:46 PM - ko1 (Koichi Sasada)

- Category set to lib

- Assignee set to seki (Masatoshi Seki)

#2 - 03/18/2012 06:46 PM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Assigned

#3 - 02/01/2015 04:01 PM - felixbuenemann (Felix Bünemann)

This issue still exists in Ruby 2.1.5.

#4 - 08/14/2019 11:53 PM - jeremyevans0 (Jeremy Evans)

- File drb-rescue-exception-5618.patch added

This issue is unrelated to whether the client defines the same exception class as the server, and the behavior in my testing appears to be the same on 1.8 as in the master branch.

The underlying issue is that the example code here uses a subclass of Exception, and DRb::DRbServer::InvokeMethod#perform does not rescue Exception. Attached is a patch that fixes this issue.

#5 - 10/16/2019 07:51 PM - jeremyevans (Jeremy Evans)

- Status changed from Assigned to Closed

Applied in changeset git|50b8033d6b7404670ee985de1473fffc10dd7d9c.

Handle subclasses of Exception in drb

This makes it so that client instances that call a method on a
server-side object that raise a subclass of Exception are handled
similarly to those that raise a subclass of StandardError.

Only reraise exceptions that we are fairly sure we don't want to
rescue.

Fixes [Bug #5618]

Files

drbserver.rb	194 Bytes	11/11/2011	larsch (Lars Christensen)
drbclient.rb	75 Bytes	11/11/2011	larsch (Lars Christensen)
drb-rescue-exception-5618.patch	1.03 KB	08/14/2019	jeremyevans0 (Jeremy Evans)