Ruby trunk - Feature #2631

Allow IO#reopen to take a block

01/23/2010 12:55 AM - djberg96 (Daniel Berger)

Status:	Assigned	
Priority:	Normal	
Assignee:	Glass_saga (Masaki Matsushita)	
Target version:		

Description

=begin
Please allow IO#reopen to accept a block. This would allow users to temporarily redirect output without having to manually reset the file descriptor. For example:

```
require 'mkmf'

# stdout redirected within block only
$stdout.reopen('/dev/null') do
if have_header('foo.h')
# Do stuff
end
end

# stdout now back to its former setting
```

I believe this is both convenient and intuitive when one considers the IO.open also takes a block.

Regards,

Dan
=end

Associated revisions

Revision f1b7003f - 06/22/2017 06:50 AM - glass

Allow IO#reopen to take a block

- io.c (rb_io_reopen): take a block and ensure the IO closed
  [Feature #2631]

- test/ruby/test_io.rb: add a test

- NEWS: add an entry for this change

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@59142 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 59142 - 06/22/2017 06:50 AM - Glass_saga (Masaki Matsushita)

Allow IO#reopen to take a block

- io.c (rb_io_reopen): take a block and ensure the IO closed
  [Feature #2631]

- test/ruby/test_io.rb: add a test

- NEWS: add an entry for this change

Revision 59142 - 06/22/2017 06:50 AM - glass

Allow IO#reopen to take a block

- io.c (rb_io_reopen): take a block and ensure the IO closed
  [Feature #2631]

- test/ruby/test_io.rb: add a test
- NEWS: add an entry for this change

Revision 59142 - 06/22/2017 06:50 AM - glass

Allow IO#reopen to take a block

- io.c (rb_io_reopen): take a block and ensure the IO closed
  [Feature #2631]
- test/ruby/test_io.rb: add a test
- NEWS: add an entry for this change

Revision 59150 - 06/23/2017 04:19 AM - Glass_saga (Masaki Matsushita)

Revert "Allow IO#reopen to take a block"

This reverts r59142.

History

#1 - 01/23/2010 01:30 AM - hongli (Hongli Lai)

=begin
I don't think this can be implemented easily. This can be implemented if the IO object in question is a File for which you know the filename. For sockets, pipes and other stuff it becomes tricker: one can emulate it backing up the underlying file descriptor by dup()ing it, but in my opinion that goes against my expectation of what #reopen should do: to replace the file descriptor with something else.
=end

#2 - 01/23/2010 04:28 AM - matz (Yukihiro Matsumoto)

Hi,

In message "Re: [ruby-core:27702] [Feature #2631] Allow IO#reopen to take a block"
on Sat, 23 Jan 2010 01:30:53 +0900, Hongli Lai redmine@ruby-lang.org writes:

> I don't think this can be implemented easily. This can be implemented if the IO object in question is a File for which you know the filename. For sockets, pipes and other stuff it becomes tricker: one can emulate it backing up the underlying file descriptor by dup()ing it, but in my opinion that goes against my expectation of what #reopen should do: to replace the file descriptor with something else.

Agreed. Besides that behavior would not be thread safe.

```
                              matz.
```

#3 - 01/23/2010 02:56 PM - shyouhei (Shyouhei Urabe)

+1, The idea of temporary intercepting any output to another IO is worth considering. Though I doubt to name that feature a "reopen".

So my suggestion is: create a new method, that takes a block, to intercept an IO.

#4 - 01/24/2010 11:07 PM - djberg96 (Daniel Berger)

On Fri, Jan 22, 2010 at 10:56 PM, Shyouhei Urabe redmine@ruby-lang.org wrote:

> Issue #2631 has been updated by Shyouhei Urabe.
>
> +1, The idea of temporary intercepting any output to another IO is worth considering. Though I doubt to name that feature a "reopen".

How about IO#redirect.

#5 - 04/02/2010 08:46 AM - znz (Kazuhiro NISHIYAMA)

- Category set to core

- Target version set to 2.0.0

#6 - 03/18/2012 02:58 PM - nahi (Hiroshi Nakamura)

- Description updated

- Assignee set to shyouhei (Shyouhei Urabe)

Method name?

#7 - 03/18/2012 03:12 PM - akr (Akira Tanaka)

I don't feel changing IO#reopen is great way to solve this issue.

How about spawn()?

Since Ruby 1.9, spawn provides a primitive for redirection.
It is thread safe.

Apart from that, I think $stdin, $stdout, $stderr can be thread-local
variables, though.

#8 - 03/18/2012 06:46 PM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Assigned

#9 - 10/27/2012 05:51 AM - shyouhei (Shyouhei Urabe)

- Target version changed from 2.0.0 to 2.6

I was poked by _ko1. But we lack a implementation proposal.

So I move its target to next minor. It might happen to be implemented some time later but not today.

Any opinions?

#10 - 06/22/2017 06:50 AM - Anonymous

- Status changed from Assigned to Closed

Applied in changeset trunk|r59142.

Allow IO#reopen to take a block

- io.c (rb_io_reopen): take a block and ensure the IO closed
  [Feature #2631]

- test/ruby/test_io.rb: add a test

- NEWS: add an entry for this change

#11 - 06/22/2017 08:35 AM - naruse (Yui NARUSE)

- Status changed from Closed to Open

□□□□□□ wrote:

> Applied in changeset trunk|r59142.
>
> ---
>
> Allow IO#reopen to take a block
>
> - io.c (rb_io_reopen): take a block and ensure the IO closed
>   [Feature #2631]
>
> - test/ruby/test_io.rb: add a test
>
> - NEWS: add an entry for this change

This patch looks doesn't recover the original $stdout.

Morever though this was approved before, this is strange because block changes global state.

If this is introduced in 7 years ago, it may be acceptable.
But in this 201x year Ruby really should support multithread unsafe code?
People who want to write unsafe code should explicitly write code like following.

```
require 'mkmf'

# stdout redirected within block only
orig_stdout = $stdout
open(IO:NULL, 'w') do |f|
  $stdout = f
  if have_header('foo.h')
    # Do stuff
  end
ensure
  $stdout = orig_stdout
end
```

#12 - 06/22/2017 08:35 AM - naruse (Yui NARUSE)

- Target version changed from 2.6 to 2.5

#13 - 06/23/2017 03:38 AM - Glass_saga (Masaki Matsushita)

- Assignee changed from shyouhei (Shyouhei Urabe) to Glass_saga (Masaki Matsushita)

- Status changed from Open to Closed

#14 - 06/23/2017 03:38 AM - Glass_saga (Masaki Matsushita)

- Status changed from Closed to Assigned

#15 - 06/23/2017 04:20 AM - Glass_saga (Masaki Matsushita)

It was thoughtless. Let me revert it.

#16 - 12/14/2017 06:23 PM - naruse (Yui NARUSE)

- Target version deleted (2.5)

#17 - 05/24/2018 01:22 PM - Bookgoogleflights (google flights)

The idea of temporary intercepting any output to another IO is worth considering. Though I doubt to name that feature a "reopen".est/ruby/test_io.rb: add a test NEWS: add an entry for this change, he new Google Flights offers all of the speed and usability of the old Google Flights, but with some great new features added on top book flights using "bookgoogleflights.wordpress.com":https://bookgoogleflights.wordpress.com/