

Ruby - Feature #12966

net/ftp to include fxp support?

11/22/2016 12:10 AM - solebox (sole box)

Status:	Rejected	
Priority:	Normal	
Assignee:	shugo (Shugo Maeda)	
Target version:		
Description		
can we implement support for fxp ? its an extension of the ftp protocol , looks like a legit thing to have no?		

History

#1 - 11/22/2016 08:57 AM - duerst (Martin Dürst)

- Assignee deleted (core)

sole box wrote:

can we implement support for fxp ? its an extension of the ftp protocol , looks like a legit thing to have no?

Probably yes. Why don't you (or somebody else) go ahead and implement it, maybe as a gem, and then we can have a look at it?

BTW, please never assign any item to "ruby-core". It will only clutter everybody's list of assigned items, without anybody feeling responsible in particular.

#2 - 11/23/2016 03:28 AM - solebox (sole box)

Martin Dürst wrote:

sole box wrote:

can we implement support for fxp ? its an extension of the ftp protocol , looks like a legit thing to have no?

Probably yes. Why don't you (or somebody else) go ahead and implement it, maybe as a gem, and then we can have a look at it?

BTW, please never assign any item to "ruby-core". It will only clutter everybody's list of assigned items, without anybody feeling responsible in particular.

ok martin,
ill implement it gladly , btw , who do i assign it to then?

#3 - 11/23/2016 07:57 AM - duerst (Martin Dürst)

sole box wrote:

Martin Dürst wrote:

BTW, please never assign any item to "ruby-core". It will only clutter everybody's list of assigned items, without anybody feeling responsible in particular.

ok martin,
ill implement it gladly

Great!

btw , who do i assign it to then?

See <https://bugs.ruby-lang.org/projects/ruby/wiki/MaintainersStdlib>.

#4 - 11/23/2016 03:00 PM - shugo (Shugo Maeda)

- Status changed from Open to Assigned
- Assignee set to shugo (Shugo Maeda)

sole box wrote:

can we implement support for fxp ? its an extension of the ftp protocol , looks like a legit thing to have no?

Do you have a specific interface in mind?

I came up with the following interface, but there may be a better interface or name.

```
class Net::FTP
  # PoC code without error check
  def self.exchange_file(dst_ftp, dst_file, src_ftp, src_file)
    host, port = dst_ftp.send(:makepasv)
    src_ftp.send(:sendport, host, port)
    th = Thread.start {
      dst_ftp.sendcmd("STOR #{dst_file}")
      dst_ftp.send(:voidresp)
    }
    src_ftp.sendcmd("RETR #{src_file}")
    src_ftp.send(:voidresp)
    th.join
  end
end

Net::FTP.open(dst_host, username: dst_user, password: dst_pass) do |dst_ftp|
  Net::FTP.open(src_host, username: src_user, password: src_pass) do |src_ftp|
    Net::FTP.exchange_file(dst_ftp, dst_file, src_ftp, src_file)
  end
end
```

It's better to support SSCN (Net::FTP#sscn), but I don't know the relationship between SSCN and RFC 4217.

Is there any written specification of SSCN?

#5 - 11/24/2016 12:38 AM - shugo (Shugo Maeda)

Shugo Maeda wrote:

```
class Net::FTP
  # PoC code without error check
  def self.exchange_file(dst_ftp, dst_file, src_ftp, src_file)
```

It might be better to swap dst_* and src_*.

#6 - 01/06/2022 01:36 AM - shugo (Shugo Maeda)

- Status changed from Assigned to Rejected

net/ftp is now a bundled gem, so I close this issue.