

Ruby master - Bug #14051

net/imap issue on idle method with an empty block

10/25/2017 03:44 AM - simayosi (SHIMAYOSHI Takao)

Status: Assigned	
Priority: Normal	
Assignee: shugo (Shugo Maeda)	
Target version:	
ruby -v:	Backport: 2.3: UNKNOWN, 2.4: UNKNOWN
Description	
Description If Net::IMAP#idle method is called with empty block, all response handlers that have added beforehand are unintentionally removed before returning from the call.	
Sample Code imap.add_response_handler do resp if resp.kind_of?(Net::IMAP::UntaggedResponse) and resp.name == "EXISTS" imap.idle_done end end imap.idle {} # the response handler firstly added is unregistered here	
Cause @response_handlers.delete(nil) is executed in Net::IMAP#remove_response_handler called from the idle method.	

History

#1 - 10/25/2017 03:49 AM - shugo (Shugo Maeda)

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

#2 - 10/25/2017 04:41 AM - shugo (Shugo Maeda)

simayosi (SHIMAYOSHI Takao) wrote:

Description

If Net::IMAP#idle method is called with empty block, all response handlers that have added beforehand are unintentionally removed before returning from the call.

Sample Code

```
imap.add_response_handler do |resp|
  if resp.kind_of?(Net::IMAP::UntaggedResponse) and resp.name == "EXISTS"
    imap.idle_done
  end
end
imap.idle {}
# the response handler firstly added is unregistered here
```

I couldn't reproduce the problem with the following code:

```
imap.add_response_handler do |resp|
  if resp.kind_of?(Net::IMAP::UntaggedResponse) and resp.name == "EXISTS"
    imap.idle_done
  end
end
p imap.response_handlers #=> [#<Proc:0x0000560887b7b958@t.rb:8>]
imap.idle {}
p imap.response_handlers #=> [#<Proc:0x0000560887b7b958@t.rb:8>]
```

Cause

@response_handlers.delete(nil) is executed in Net::IMAP#remove_response_handler called from the idle method.

Even if the body of a block is empty, a block parameter cannot be nil.

```
def foo(&b)
  p b
end
foo {} #=> #<Proc:0x00005595c4e7f298@(irb):2>
```

And even if no block is given, a LocalJumpError will be raised.

```
def idle(timeout = nil, &response_handler)
  raise LocalJumpError, "no block given" unless response_handler
end
```