Ruby master - Bug #11523

optparse short options will match complete options

09/11/2015 04:03 PM - mjrk (Micha J)

Status:	Assigned		
Priority:	Normal		
Assignee:	nobu (Nobuyoshi Nakada)		
Target version:			
ruby -v:	2.2	**Backport:**	2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN

Description

In short, if I define an option like "-F", "--irs [OCTAL]", -i will match this option, although the short version is defined as -F.

In long, this can be quite troublesome:

See the provided example

http://ruby-doc.org/stdlib-2.2.0/libdoc/optparse/rdoc/OptionParser.html

and change or remove the "-i", "--inplace [EXTENSION]" option for something else than i:

Now, the -i will still match, but the other option "-F", "--irs [OCTAL]"!

In a more complete stack this resulted in a hard to find error. Also, to fix this (and raise the required error) you need to check the ARGV directly which renders optparse a bit less useful.

History

#1 - 09/19/2015 09:58 AM - mjrk (Micha J)

Modified code example optiontest.rb

```
require 'optparse'
require 'pp'

class OptparseExample
  def self.parse(args)
    options = {}

    opt_parser = OptionParser.new do |opts|
      opts.on("-F", "--irs [OCTAL]", OptionParser::OctalInteger,
              "Specify record separator (default \\0)") do |rs|
        options["trigger"] = "I got triggered"
      end
    end

    opt_parser.parse!(args)
    options
  end  # parse()

end  # class OptparseExample

options = OptparseExample.parse(ARGV)
pp options
pp ARGV

$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 15.04
Release:    15.04
Codename:   vivid
$ ruby -v
ruby 2.2.1p85 (2015-02-26 revision 49769) [x86_64-linux]
$ ruby optiontest.rb -i
{"trigger"=>"I got triggered"}
```

```
[]
```

The -F, --irs option is triggered with -i. Especially confusing as the short version is specified as -F

#2 - 09/24/2015 04:49 AM - shishir127 (Shishir Joshi)

Try executing this script. The only difference between the previous script and this one is that instead of returning the options variable, opt_parser.parse!(args) is being returned from the method. I did not get the error with this script.

```
require 'optparse'
require 'pp'

class OptparseExample
  def self.parse(args)
    options = {}

    opt_parser = OptionParser.new do |opts|
      opts.on("-F", "--irs [OCTAL]", OptionParser::OctalInteger,
              "Specify record separator (default \\0)") do |rs|
        options["trigger"] = "I got triggered"
      end
    end

    opt_parser.parse!(args)
  end  # parse()

end  # class OptparseExample

options = OptparseExample.parse(ARGV)
pp options
pp ARGV
```

#3 - 02/12/2016 06:13 PM - vo.x (Vit Ondruch)

I believe that:

1. This is what OptParser doing by default. If your code were opts.on("--irs [OCTAL]", OptionParser::OctalInteger,, i.e. without the '-F', you would get '--irs' as well as shorthand '-i' available. Your case is bit specific, that the short and long parameters are quite different.
2. If you override the '-i' option somewhere and give it different meaning, the OptParser will handle it correctly.
3. If you really want to avoid the default short version of '-i', I can't see anything easier then fire the OptionParser::InvalidOption in the '-i' handler.

#4 - 07/01/2019 10:24 PM - jeremyevans0 (Jeremy Evans)

- Assignee set to nobu (Nobuyoshi Nakada)

- Status changed from Open to Assigned

- File optparse-require-exact.patch added

I'm not sure I would consider this a bug, but I can definitely see it as an undesired feature in some cases. I think it would be worthwhile to add a way to turn off this feature, and require that options specified on the command line match exactly. Attached is a patch that adds a require_exact accessor that does this.

Files

optparse-require-exact.patch	3.06 KB	07/01/2019	jeremyevans0 (Jeremy Evans)