

Ruby - Bug #12100

CSV converters fail when using single arg proc

02/22/2016 04:14 PM - mdaubert (Matthew Daubert)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	all versions 2.0, 2.1, 2.2, 2.3, trunk	Backport: 2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN

Description

CSV#parse (and others) throw an ArgumentError when passing a single argument Proc as a converter because of a performance optimization check that assumes Proc#arity is always positive. Lambdas with one argument work fine as do Procs and lambdas with two arguments. The documentation suggests to me that either should work. Supporting the Proc variant is trivial and allows to_proc and to_proc shortcut syntax to be used eliminating the surprising behavior and uninformative error message. Illustration of the problem is below and a patch with tests is attached. I'm also creating a PR on Github because I think this is a minor enough change to be merged quickly. Thank you.

```
require 'csv'
puts CSV.parse("  foo  ,  bar  ", converters: -> f { f.strip }).inspect
puts CSV.parse("  foo  ,  bar  ", converters: :strip.to_proc).inspect

# Without patch this outputs:
# [{"foo", "bar"}]
# /home/mdaubert/.rbenv/versions/2.3.0/lib/ruby/2.3.0/csv.rb:2205:in `strip': wrong number of arguments (given 1, expected 0) (ArgumentError)

# With patch this outputs:
# [{"foo", "bar"}]
# [{"foo", "bar"}]
```

History

#1 - 02/22/2016 04:41 PM - recursive-madman (Recursive Madman)

I'm pretty sure this will introduce new bugs.
The meaning of an arity of -1 is "any number of arguments".

In reality you're not passing a proc which takes 1 argument (which would work w/o your patch), but a proc which takes any number of arguments. The reason for the :strip.to_proc Proc to take an arbitrary number of arguments (while String#strip) does not, is that Symbol#to_proc cannot know the arity of the method call it is wrapping, thus always accepting any number of arguments.

To illustrate,

:strip.to_proc is equivalent to -> receiver, *rest { receiver.strip(*rest) }

When you pass a proc like:
-> receiver { receiver.strip }
directly, everything should work fine.

Only by using Symbol#to_proc you "break" the arity check within the CSV library.

#2 - 02/22/2016 04:44 PM - mdaubert (Matthew Daubert)

Upon further investigation it looks like this might actually stem from Symbol#to_proc returning a Proc with unspecified number of arguments, which makes sense, but I think the fix here is valid regardless. What REcursive Madman said... thanks!

#3 - 02/22/2016 05:22 PM - mdaubert (Matthew Daubert)

Thanks for the explanation Madman, that's super helpful. So the actual bug is reproduced when passing a proc/non-lambda with any number of arguments but I agree, this would cause regressions in a case like this...

```
parser.convert do |*args|
  field, info = *args
  # do things and return
end
```

<https://bugs.ruby-lang.org/issues/12100#>
I don't think there's a way to fix this without causing a regression so this can be closed.

#4 - 03/03/2016 05:41 AM - shyouhei (Shyouhei Urabe)
- *Status changed from Open to Closed*

closing as per request

#5 - 03/24/2016 07:16 AM - usa (Usaku NAKAMURA)
- *Status changed from Closed to Rejected*

Files

fix_csv_converter_single_arg_proc.patch	1.78 KB	02/22/2016	mdaubert (Matthew Daubert)
---	---------	------------	----------------------------