

Ruby trunk - Feature #9451

Refinements and unary & (to_proc)

01/26/2014 01:57 PM - rbjl (Jan Lejis)

Status:	Closed	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	
Target version:		
Description		
Not sure if this is a bug or feature request:		
<pre>require 'minitest/autorun' require 'set' module ClassToProc refine Class do def to_proc lambda{ *args self.new(*args) } end end end using ClassToProc describe 'Class#to_proc' do it 'works when called directly' do Set.to_proc[[1,2]].must_equal Set[1,2] end it 'fails when called via ampersand' do [[1,2]].map(&Set).must_equal [Set[1,2]] end end</pre>		
The second example errors with <i>NoMethodError: super: no superclass method `to_proc' for Set:Class</i>		
Would be great to have it, though.		
Related issues:		
Has duplicate Ruby trunk - Feature #12638: Symbol#to_proc probable bug (seems...		Closed
Has duplicate Ruby trunk - Feature #12079: Loosening the condition for refine...		Rejected

Associated revisions

Revision 35a29390 - 10/15/2016 05:17 AM - nobu (Nobuyoshi Nakada)

vm_args.c: allow refinements in Symbol proc

- vm_args.c (refine_sym_proc_call): search and call method with refinements.
- vm_args.c (vm_caller_setup_arg_block): enable refinements when enabled in the caller. [Feature #9451]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@56426 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 56426 - 10/15/2016 05:17 AM - nobu (Nobuyoshi Nakada)

vm_args.c: allow refinements in Symbol proc

- vm_args.c (refine_sym_proc_call): search and call method with refinements.
- vm_args.c (vm_caller_setup_arg_block): enable refinements when enabled in the caller. [Feature #9451]

Revision 56426 - 10/15/2016 05:17 AM - nobu (Nobuyoshi Nakada)

vm_args.c: allow refinements in Symbol proc

- vm_args.c (refine_sym_proc_call): search and call method with refinements.
- vm_args.c (vm_caller_setup_arg_block): enable refinements when enabled in the caller. [Feature #9451]

Revision 56426 - 10/15/2016 05:17 AM - nobu (Nobuyoshi Nakada)

vm_args.c: allow refinements in Symbol proc

- vm_args.c (refine_sym_proc_call): search and call method with refinements.
- vm_args.c (vm_caller_setup_arg_block): enable refinements when enabled in the caller. [Feature #9451]

Revision 56426 - 10/15/2016 05:17 AM - nobu (Nobuyoshi Nakada)

vm_args.c: allow refinements in Symbol proc

- vm_args.c (refine_sym_proc_call): search and call method with refinements.
- vm_args.c (vm_caller_setup_arg_block): enable refinements when enabled in the caller. [Feature #9451]

Revision 65daf7be - 10/15/2016 09:05 AM - nobu (Nobuyoshi Nakada)

vm_args.c: cache procs

- vm_args.c (vm_caller_setup_arg_block): cache proc with refinements. [Feature #9451]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@56427 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 56427 - 10/15/2016 09:05 AM - nobu (Nobuyoshi Nakada)

vm_args.c: cache procs

- vm_args.c (vm_caller_setup_arg_block): cache proc with refinements. [Feature #9451]

Revision 56427 - 10/15/2016 09:05 AM - nobu (Nobuyoshi Nakada)

vm_args.c: cache procs

- vm_args.c (vm_caller_setup_arg_block): cache proc with refinements. [Feature #9451]

Revision 56427 - 10/15/2016 09:05 AM - nobu (Nobuyoshi Nakada)

vm_args.c: cache procs

- vm_args.c (vm_caller_setup_arg_block): cache proc with refinements. [Feature #9451]

Revision 56427 - 10/15/2016 09:05 AM - nobu (Nobuyoshi Nakada)

vm_args.c: cache procs

- vm_args.c (vm_caller_setup_arg_block): cache proc with refinements. [Feature #9451]

Revision 9ec940a4 - 10/15/2016 10:59 PM - nobu (Nobuyoshi Nakada)

test_refinement.rb: test_symbol_proc

- test/ruby/test_refinement.rb (test_symbol_proc): move from test_symbol.rb [Feature #9451]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@56431 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 56431 - 10/15/2016 10:59 PM - nobu (Nobuyoshi Nakada)

test_refinement.rb: test_symbol_proc

- test/ruby/test_refinement.rb (test_symbol_proc): move from test_symbol.rb [Feature #9451]

Revision 56431 - 10/15/2016 10:59 PM - nobu (Nobuyoshi Nakada)

test_refinement.rb: test_symbol_proc

- test/ruby/test_refinement.rb (test_symbol_proc): move from test_symbol.rb [Feature #9451]

Revision 56431 - 10/15/2016 10:59 PM - nobu (Nobuyoshi Nakada)

test_refinement.rb: test_symbol_proc

- test/ruby/test_refinement.rb (test_symbol_proc): move from test_symbol.rb [Feature #9451]

Revision 56431 - 10/15/2016 10:59 PM - nobu (Nobuyoshi Nakada)

test_refinement.rb: test_symbol_proc

- test/ruby/test_refinement.rb (test_symbol_proc): move from test_symbol.rb [Feature #9451]

Revision 9be1910a - 10/23/2016 02:03 AM - nobu (Nobuyoshi Nakada)

test_refinement.rb: Symbol#to_proc test

- test/ruby/test_refinement.rb (test_symbol_proc): a Symbol is converted to a Proc in the caller's context. [Feature #9451]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@56476 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 56476 - 10/23/2016 02:03 AM - nobu (Nobuyoshi Nakada)

test_refinement.rb: Symbol#to_proc test

- test/ruby/test_refinement.rb (test_symbol_proc): a Symbol is converted to a Proc in the caller's context. [Feature #9451]

Revision 56476 - 10/23/2016 02:03 AM - nobu (Nobuyoshi Nakada)

test_refinement.rb: Symbol#to_proc test

- test/ruby/test_refinement.rb (test_symbol_proc): a Symbol is converted to a Proc in the caller's context. [Feature #9451]

Revision 56476 - 10/23/2016 02:03 AM - nobu (Nobuyoshi Nakada)

test_refinement.rb: Symbol#to_proc test

- test/ruby/test_refinement.rb (test_symbol_proc): a Symbol is converted to a Proc in the caller's context. [Feature #9451]

Revision 56476 - 10/23/2016 02:03 AM - nobu (Nobuyoshi Nakada)

test_refinement.rb: Symbol#to_proc test

- test/ruby/test_refinement.rb (test_symbol_proc): a Symbol is converted to a Proc in the caller's context. [Feature #9451]

History

#1 - 01/29/2014 12:50 AM - matz (Yukihiro Matsumoto)

- Assignee set to shugo (Shugo Maeda)

As #=== from case statement and #each from for statement, I think refinement should be effective to methods that called implicitly. Let us discuss about the issue.

Matz.

#2 - 01/29/2014 04:05 AM - shugo (Shugo Maeda)

- Tracker changed from Bug to Feature

Yukihiro Matsumoto wrote:

As #=== from case statement and #each from for statement, I think refinement should be effective to methods that called implicitly. Let us discuss about the issue.

What do you think of other implicit conversions such as to_a called by the splat operator.

#3 - 02/01/2014 11:21 PM - shugo (Shugo Maeda)

- Category set to core

- Status changed from Open to Assigned

- Assignee changed from shugo (Shugo Maeda) to matz (Yukihiro Matsumoto)

- Target version set to 2.2.0

I'd like to clarify the design policy rather than adding features ad hoc.

All implicit method calls in syntactic constructs should be affected by refinements, right?
Do other implementers, especially JRuby people, accept this policy?

#4 - 05/30/2014 02:46 PM - nobu (Nobuyoshi Nakada)

- Description updated

IMHO, splat should be affected by refinements, because there is no intermediate method calls.

#5 - 12/06/2015 03:13 AM - justcolin (Colin Fulton)

Is there any update on this feature? In Ruby 2.2.3 I still run into a problem where the unary & can not be used when the method was added with a refinement, such as the following:

```
module Example
  refine String do
    def pugs
      "Pugs!"
    end
  end
end

using Example

('a'..'z').map(&pugs)
```

I can understand why they currently do not work, but it leads to uglier code.

Cheers!

#6 - 12/11/2015 06:42 PM - justcolin (Colin Fulton)

Sorry, just saw that [issue 9786](#) is equivalent to this one and was marked as rejected. Thanks!

#7 - 07/30/2016 01:18 AM - nobu (Nobuyoshi Nakada)

- Has duplicate Feature #12638: *Symbol#to_proc probable bug (seems not to support refinements) added*

#8 - 09/07/2016 07:29 AM - matz (Yukihiro Matsumoto)

I now think it's OK to accept this proposal. But I don't know how difficult to implement the issue.

Matz.

#9 - 09/08/2016 06:49 AM - shugo (Shugo Maeda)

- Has duplicate Feature #12079: *Loosening the condition for refinement added*

#10 - 10/15/2016 05:17 AM - nobu (Nobuyoshi Nakada)

- Status changed from Assigned to Closed

Applied in changeset [r56426](#).

vm_args.c: allow refinements in Symbol proc

- vm_args.c (refine_sym_proc_call): search and call method with refinements.
- vm_args.c (vm_caller_setup_arg_block): enable refinements when enabled in the caller. [Feature [#9451](#)]