

Ruby trunk - Feature #9783

[PATCH] Add Method#curry

04/29/2014 05:39 AM - plexus (Arne Brasseur)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	
Description	
There is already Proc#curry, but to curry a method you need to go through to_proc. This patch adds Method#curry which delegates to method.to_proc.curry.	
Looking forward to seeing this discussed.	

Associated revisions

Revision b2066b11 - 06/19/2014 04:42 AM - nobu (Nobuyoshi Nakada)

proc.c: Implement Method#curry

- proc.c (rb_method_curry): Implement Method#curry, which delegates to to_proc.curry. [ruby-core:62212] [Feature #9783]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@46461 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 46461 - 06/19/2014 04:42 AM - nobu (Nobuyoshi Nakada)

proc.c: Implement Method#curry

- proc.c (rb_method_curry): Implement Method#curry, which delegates to to_proc.curry. [ruby-core:62212] [Feature #9783]

Revision 46461 - 06/19/2014 04:42 AM - nobu (Nobuyoshi Nakada)

proc.c: Implement Method#curry

- proc.c (rb_method_curry): Implement Method#curry, which delegates to to_proc.curry. [ruby-core:62212] [Feature #9783]

Revision 46461 - 06/19/2014 04:42 AM - nobu (Nobuyoshi Nakada)

proc.c: Implement Method#curry

- proc.c (rb_method_curry): Implement Method#curry, which delegates to to_proc.curry. [ruby-core:62212] [Feature #9783]

Revision 46461 - 06/19/2014 04:42 AM - nobu (Nobuyoshi Nakada)

proc.c: Implement Method#curry

- proc.c (rb_method_curry): Implement Method#curry, which delegates to to_proc.curry. [ruby-core:62212] [Feature #9783]

Revision 46461 - 06/19/2014 04:42 AM - nobu (Nobuyoshi Nakada)

proc.c: Implement Method#curry

- proc.c (rb_method_curry): Implement Method#curry, which delegates to to_proc.curry. [ruby-core:62212] [Feature #9783]

Revision 46461 - 06/19/2014 04:42 AM - nobu (Nobuyoshi Nakada)

proc.c: Implement Method#curry

- proc.c (rb_method_curry): Implement Method#curry, which delegates to to_proc.curry. [ruby-core:62212] [Feature #9783]

History

#1 - 04/29/2014 07:29 AM - plexus (Arne Brasseur)

- File 0001-Implement-Method-curry-which-delegates-to-to_proc.cu.patch added

Updated the patch with documentation. There's an interesting side effect to be considered. Method#to_proc returns a lambda that checks its arity. Proc#curry takes an optional arity argument. If this doesn't match the method's arity then the result can not be called.

```

def two_args(a,b)
end

curried = method(:two_args).to_proc.curry(1)

curried.(1)
# -:4:in `curry': wrong number of arguments (1 for 2) (ArgumentError)
#   from -:4:in `'

curried.(1, 2)
# -:4:in `curry': wrong number of arguments (1 for 2) (ArgumentError)
#   from -:4:in `'

curried.(1, 2, 3)
# -:4:in `curry': wrong number of arguments (1 for 2) (ArgumentError)
#   from -:4:in `'

```

In the case of a method with variable arguments the arity argument is important though, because otherwise the currying has no effect.

```

def varargs(*args)
  args
end

curried = method(:varargs).to_proc.curry
curried.(1) # => [1]
curried.(1,2) # => [1, 2]
curried = method(:varargs).to_proc.curry(3)
curried.(1) # => #<Proc:0x007f0cb8fc6fe0 (lambda)>
curried.(1).(2).(3) # => [1, 2, 3]

```

I think it would make sense for a Method#curry method to raise an error or at least give a warning when passing the wrong arity to a fixed-arity function.

#2 - 06/18/2014 11:19 PM - matz (Yukihiro Matsumoto)

Looks good to me, except that I don't see any real use-case. But it's true for Proc#curry itself.

Matz.

#3 - 06/19/2014 04:42 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed
 - % Done changed from 0 to 100

Applied in changeset r46461.

proc.c: Implement Method#curry

- proc.c (rb_method_curry): Implement Method#curry, which delegates to to_proc.curry. [ruby-core:62212] [Feature [#9783](#)]

Files

0001-Implement-Method-curry-which-delegates-to-to_proc.cu.patch	1.86 KB	04/29/2014	plexus (Arne Brasseur)
0001-Implement-Method-curry-which-delegates-to-to_proc.cu.patch	3.19 KB	04/29/2014	plexus (Arne Brasseur)