

## Ruby trunk - Feature #7883

### Add Regexp#to\_proc

02/19/2013 04:58 PM - rklemme (Robert Klemme)

<b>Status:</b>	Open
<b>Priority:</b>	Normal
<b>Assignee:</b>	matz (Yukihiro Matsumoto)
<b>Target version:</b>	
<b>Description</b>	
Just a small addition to the standard library:	
<pre>class Regexp   def to_proc; lambda { s  self =~ s} end end</pre>	
With that one can use a Regexp everywhere a Proc is used as filtering criteria saving a bit of typing. While we have Enumerable#grep already there may be other cases where you want to do something like	
<pre>irb(main):008:0&gt; %w{foo bar baz}.select &amp;/\Ab/ =&gt; ["bar", "baz"] irb(main):009:0&gt; %w{foo bar baz}.reject &amp;/\Ab/ =&gt; ["foo"] irb(main):010:0&gt; %w{foo bar baz}.find &amp;/\Ab/ =&gt; "bar"</pre>	
Note: line 9 and 10 are not possible with Enumerable#grep AFAIK.	
I see low risk of breaking something.	

### History

#### #1 - 02/19/2013 07:24 PM - nobu (Nobuyoshi Nakada)

You can use `^Ab/.method(:=~)`.

#### #2 - 02/19/2013 07:41 PM - charliesome (Charlie Somerville)

You can use `^Ab/.method(:=~)`.

This is more typing than just using `'{|x| x =~ ^Ab/}'`.

I like this idea, it's something I have to do quite often.

I've also wanted a `'grep_v'` method before, but `reject` with `Regexp#to_proc` would suit perfectly.

#### #3 - 02/19/2013 07:53 PM - regularfry (Alex Young)

On 19/02/2013 07:58, rklemme (Robert Klemme) wrote:

Issue [#7883](#) has been reported by rklemme (Robert Klemme).

Lovely. Definite +1 from me (for whatever that's worth).

--

Alex

#### #4 - 02/19/2013 08:47 PM - rklemme (Robert Klemme)

nobu (Nobuyoshi Nakada) wrote:

You can use `^Ab/.method(:=~)`.

Yes, but that's not the point. In that case I'd prefer the real block as Charlie suggested.

#### #5 - 02/19/2013 11:06 PM - prijutme4ty (Ilya Vorontsov)

Cool idea. +1

#### #6 - 02/19/2013 11:23 PM - judofyr (Magnus Holm)

```
class Regexp
  def to_proc; lambda {|s| self =~ s} end
end
```

A more general case:

```
class Object
  def to_proc
    lambda { |other| self === other }
  end
end
```

(Not that I think this is going to be accepted...)

#### #7 - 02/20/2013 01:58 AM - injekt (Lee Jarvis)

I like this, and I especially like the modified version from judofyr. You could also do ["abc", 3, "foo", etc].find(&Integer) which is neat, but maybe that's a push in the wrong direction, I'm not sure.

#### #8 - 02/21/2013 10:36 PM - goshakkk (Gosha Arinich)

Nice one, +1 on it as well.

#### #9 - 02/21/2013 11:23 PM - jjyr (Jinyang Jiang)

judofyr (Magnus Holm) wrote:

```
class Regexp
  def to_proc; lambda {|s| self =~ s} end
end
```

A more general case:

```
class Object
  def to_proc
    lambda { |other| self === other }
  end
end
```

(Not that I think this is going to be accepted...)

cool!

#### #10 - 02/22/2013 09:04 AM - ko1 (Koichi Sasada)

- Assignee set to matz (Yukihiro Matsumoto)

- Target version changed from 1.9.3 to 2.6

#### #11 - 12/23/2014 12:01 AM - rklemme (Robert Klemme)

Was any of the two suggested forms ever realized?

#### #12 - 12/23/2014 03:09 AM - nobu (Nobuyoshi Nakada)

- Description updated

Nothing, and line 10 is possible with grep

```
%w{foo bar baz}.grep(/\Ab/) {|s|break s}
```

Anyway, a patch.

```

diff --git i/re.c w/re.c
index 1a0f0b1..d345f17 100644
--- i/re.c
+++ w/re.c
@@ -801,6 +801,19 @@ rb_reg_named_captures(VALUE re)
     return hash;
 }

+static VALUE
+rb_reg_to_proc(VALUE re)
+{
+  const ID id = 1;
+  VALUE proc = rb_attr_get(re, id);
+  if (NIL_P(proc)) {
+    proc = rb_obj_method(re, ID2SYM(rb_intern("~")));
+    proc = rb_funcall(proc, rb_intern("to_proc"), 0, 0);
+    rb_ivar_set(re, id, proc);
+  }
+  return proc;
+}
+
+static int
+onig_new_with_source(regex_t** reg, const UChar* pattern, const UChar* pattern_end,
+  OnigOptionType option, OnigEncoding enc, const OnigSyntaxType* syntax,
@@ -3660,6 +3673,7 @@ Init_Regexp(void)
   rb_define_method(rb_cRegexp, "fixed_encoding?", rb_reg_fixed_encoding_p, 0);
   rb_define_method(rb_cRegexp, "names", rb_reg_names, 0);
   rb_define_method(rb_cRegexp, "named_captures", rb_reg_named_captures, 0);
+  rb_define_method(rb_cRegexp, "to_proc", rb_reg_to_proc, 0);

   /* see Regexp.options and Regexp.new */
   rb_define_const(rb_cRegexp, "IGNORECASE", INT2FIX(ONIG_OPTION_IGNORECASE));

```

### #13 - 12/23/2014 01:15 PM - rklemme (Robert Klemme)

Nobuyoshi Nakada wrote:

Nothing, and line 10 is possible with grep

```
%w{foo bar baz}.grep(/\Ab/) {|s|break s}
```

Yes, but a bit lengthy.

Anyway, a patch.

Cool! Thanks a lot!

### #14 - 12/24/2014 01:17 AM - sawa (Tsuyoshi Sawada)

If the proposed patch to issue [#9602](#) gets accepted, then that will cover this issue in a more general way.

### #15 - 12/25/2017 06:15 PM - naruse (Yui NARUSE)

- Target version deleted (2.6)