

Ruby - Bug #16450

Range#min with a beginless range should raise an explicit exception

12/25/2019 04:33 AM - mame (Yusuke Endoh)

Status:	Closed	Backport: 2.5: UNKNOWN, 2.6: UNKNOWN
Priority:	Normal	
Assignee:	mame (Yusuke Endoh)	
Target version:		
ruby -v:	ruby 2.7.0rc2 (2019-12-22 master 75acbd5f00) [x86_64-linux]	
Description		
Range#max with an endless range raises an explicit exception, so Range#min with a beginless range do the same.		
<pre>\$ ruby -e '(0..).max' Traceback (most recent call last): 1: from -e:1:in `<main>' -e:1:in `max': cannot get the maximum of endless range (RangeError)</pre>		
<pre>\$ ruby -e '(..0).min' Traceback (most recent call last): 1: from -e:1:in `<main>' -e:1:in `min': comparison of NilClass with 0 failed (ArgumentError)</pre>		
Range#max with a block has a similar issue.		
<pre>\$ ruby -e '(0..).min { a,b a }' Traceback (most recent call last): 1: from -e:1:in `<main>' -e:1:in `min': cannot get the minimum of endless range with custom comparison method (RangeError)</pre>		
<pre>\$ ruby -e '(..0).max { a,b a }' Traceback (most recent call last): 3: from -e:1:in `<main>'   2: from -e:1:in `max' 1: from -e:1:in `max' -e:1:in `each': can't iterate from NilClass (TypeError)</pre>		
<pre>diff --git a/range.c b/range.c index fe956197c6..bf14c0c7a7 100644 --- a/range.c +++ b/range.c @@ -1136,6 +1136,10 @@ range_last(int argc, VALUE *argv, VALUE range) static VALUE range_min(int argc, VALUE *argv, VALUE range) { + if (NIL_P(RANGE_BEG(range))) { + rb_raise(rb_eRangeError, "cannot get the minimum of beginless range"); + } + if (rb_block_given_p()) { if (NIL_P(RANGE_END(range))) { rb_raise(rb_eRangeError, "cannot get the minimum of endless range with custom comparison method"); @@ -1185,6 +1189,9 @@ range_max(int argc, VALUE *argv, VALUE range) } if (rb_block_given_p() (EXCL(range) && !nm) argc) { + if (NIL_P(RANGE_BEG(range))) { + rb_raise(rb_eRangeError, "cannot get the maximum of beginless range with custom comparison method"); + } return rb_call_super(argc, argv);</pre>		

```

    }
    else {
diff --git a/test/ruby/test_range.rb b/test/ruby/test_range.rb
index 800cee92cc..b37dbbc433 100644
--- a/test/ruby/test_range.rb
+++ b/test/ruby/test_range.rb
@@ -81,6 +81,8 @@ def test_min
    assert_equal(nil, (2..1).min)
    assert_equal(1, (1...2).min)
    assert_equal(1, (1..).min)
+   assert_raise(RangeError) { (..1).min }
+   assert_raise(RangeError) { (...1).min }

    assert_equal(1.0, (1.0..2.0).min)
    assert_equal(nil, (2.0..1.0).min)
@@ -93,6 +95,8 @@ def test_min
    assert_equal([0,1,2], (0..10).min(3))
    assert_equal([0,1], (0..1).min(3))
    assert_equal([0,1,2], (0..).min(3))
+   assert_raise(RangeError) { (..1).min(3) }
+   assert_raise(RangeError) { (...1).min(3) }

    assert_raise(RangeError) { (0..).min {|a, b| a <=> b } }
  end
@@ -119,6 +123,8 @@ def test_max
    assert_equal([9,8,7], (0...10).max(3))
    assert_raise(RangeError) { (1..).max(3) }
    assert_raise(RangeError) { (1...).max(3) }
+
+   assert_raise(RangeError) { (..0).min {|a, b| a <=> b } }
  end

  def test_minmax

```

Associated revisions

Revision 81e377023c490998a3fec245ca2fb2b3c710c2c6 - 12/25/2019 04:36 AM - mame (Yusuke Endoh)

range.c: Range#min with a beginless one now raise an explicit exception

[Bug #16450]

Revision 81e377023c490998a3fec245ca2fb2b3c710c2c6 - 12/25/2019 04:36 AM - mame (Yusuke Endoh)

range.c: Range#min with a beginless one now raise an explicit exception

[Bug #16450]

Revision 81e37702 - 12/25/2019 04:36 AM - mame (Yusuke Endoh)

range.c: Range#min with a beginless one now raise an explicit exception

[Bug #16450]

History

#1 - 12/26/2019 05:39 AM - mame (Yusuke Endoh)

- Status changed from Assigned to Closed

81e377023c490998a3fec245ca2fb2b3c710c2c6