

Ruby master - Bug #14845

Endless Range with nil

06/13/2018 07:53 AM - igaiga (Kuniaki IGARASHI)

Status: Closed	
Priority: Normal	
Assignee: matz (Yukihiro Matsumoto)	
Target version:	
ruby -v: 2.6.0preview2	Backport: 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN
Description endless Range <ul style="list-style-type: none">1..nil endless Rangeendless Range to_a1.. to_a endless Range to_a Ruby 2.5.1 1..nil ArgumentError (bad value for range) 1..nil endless Range Info <ul style="list-style-type: none">An endless range (1..) : https://bugs.ruby-lang.org/issues/12912twitter : https://twitter.com/igaiga555/status/1006715631796813824	
Related issues: Related to Ruby master - Misc #17637: Endless ranges with `nil` boundary weir... Open	

Associated revisions

Revision 48de2ea5 - 06/13/2018 09:04 AM - mame (Yusuke Endoh)

range.c: prohibit (1..nil)

Now endless range can be created by either a literal (1..) or explicit range creation Range.new(1, nil). [Bug #14845]

This change is intended for "early failure"; for example, (1..var).to_a causes out of memory if var is inadvertently nil.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@63646 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 63646 - 06/13/2018 09:04 AM - mame (Yusuke Endoh)

range.c: prohibit (1..nil)

Now endless range can be created by either a literal (1..) or explicit range creation Range.new(1, nil). [Bug #14845]

This change is intended for "early failure"; for example, (1..var).to_a causes out of memory if var is inadvertently nil.

Revision 63646 - 06/13/2018 09:04 AM - mame (Yusuke Endoh)

range.c: prohibit (1..nil)

Now endless range can be created by either a literal (1..) or explicit range creation Range.new(1, nil). [Bug #14845]

This change is intended for "early failure"; for example, (1..var).to_a causes out of memory if var is inadvertently nil.

Revision 606d6b34 - 06/13/2018 11:00 AM - mame (Yusuke Endoh)

Revert "range.c: prohibit (1..nil)"

This reverts commit a44c010764a16ae09aaed49d76eec055ca0057c8.
Refs #14845.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@63649 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 63649 - 06/13/2018 11:00 AM - mame (Yusuke Endoh)

Revert "range.c: prohibit (1..nil)"

This reverts commit a44c010764a16ae09aaed49d76eec055ca0057c8.
Refs #14845.

Revision 63649 - 06/13/2018 11:00 AM - mame (Yusuke Endoh)

Revert "range.c: prohibit (1..nil)"

This reverts commit a44c010764a16ae09aaed49d76eec055ca0057c8.
Refs #14845.

Revision c19ecf05 - 06/22/2018 02:58 AM - mame (Yusuke Endoh)

range.c: Range#to_a now raises RangeError if it is endless

Fixes [Bug #14845]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@63714 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 63714 - 06/22/2018 02:58 AM - mame (Yusuke Endoh)

range.c: Range#to_a now raises RangeError if it is endless

Fixes [Bug #14845]

Revision 63714 - 06/22/2018 02:58 AM - mame (Yusuke Endoh)

range.c: Range#to_a now raises RangeError if it is endless

Fixes [Bug #14845]

History

#1 - 06/13/2018 08:06 AM - mame (Yusuke Endoh)

early failure

SQL DSL count: (min..max) Range max = nil (1..nil) count: max ? (min..max) : (min..) DSL

to_a (1..nil).to_a enum.cycle.to_a to_a

(1..nil)

#2 - 06/13/2018 09:05 AM - mame (Yusuke Endoh)

- Status changed from Open to Closed

Applied in changeset [trunk|r63646](#).

range.c: prohibit (1..nil)

Now endless range can be created by either a literal (1..) or explicit range creation Range.new(1, nil). [Bug #14845]

This change is intended for "early failure"; for example, (1..var).to_a causes out of memory if var is inadvertently nil.

#3 - 06/13/2018 09:06 AM - mame (Yusuke Endoh)

- Assignee set to matz (Yukihiko Matsumoto)

- Status changed from Closed to Assigned

https://twitter.com/yukihiko_matz/status/1006737384820445184

Files

range_to_a.patch

1.56 KB

06/13/2018

mame (Yusuke Endoh)