

## Ruby trunk - Bug #7539

### Misleading error message "can't convert nil into string"

12/10/2012 09:29 PM - connec (Chris Connelly)

<b>Status:</b> Closed	
<b>Priority:</b> Normal	
<b>Assignee:</b> marcandre (Marc-Andre Lafortune)	
<b>Target version:</b> 2.6	
<b>ruby -v:</b> ruby 1.9.3p0 (2011-10-30 revision 33570) [x86_64-linux]	<b>Backport:</b>
<b>Description</b>	
<p>When trying to call <code>String#+</code> with <code>nil</code> as an argument, you get the error "can't convert nil into String", which does not make sense (in fact seeming blatantly false) as <code>nil.to_s</code>, <code>String(nil)</code> etc. all return "" without errors.</p> <p>Ideally, this method should use <code>to_s</code> to convert the argument, or else report an error along the lines of "can't append nil to string".</p> <p>Minimal test case:</p> <p>Actual:</p> <pre>&gt; "" + nil TypeError: can't convert nil into String</pre> <p>Expected:</p> <pre>&gt; "" + nil "</pre>	

#### Associated revisions

##### Revision c34e9f23 - 01/29/2013 10:00 PM - marcandre (Marc-Andre Lafortune)

- object.c: Improve error for failed implicit conversions [Bug #7539]
- error.c: Adapt rdoc
- test/ruby/test\_object.rb: Test for above

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@38979 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 38979 - 01/29/2013 10:00 PM - marcandre (Marc-Andre Lafortune)

- object.c: Improve error for failed implicit conversions [Bug #7539]
- error.c: Adapt rdoc
- test/ruby/test\_object.rb: Test for above

##### Revision 38979 - 01/29/2013 10:00 PM - marcandre (Marc-Andre Lafortune)

- object.c: Improve error for failed implicit conversions [Bug #7539]
- error.c: Adapt rdoc
- test/ruby/test\_object.rb: Test for above

**Revision 38979 - 01/29/2013 10:00 PM - marcandre (Marc-Andre Lafortune)**

- object.c: Improve error for failed implicit conversions [Bug #7539]
- error.c: Adapt rdoc
- test/ruby/test\_object.rb: Test for above

**Revision 38979 - 01/29/2013 10:00 PM - marcandre (Marc-Andre Lafortune)**

- object.c: Improve error for failed implicit conversions [Bug #7539]
- error.c: Adapt rdoc
- test/ruby/test\_object.rb: Test for above

**Revision 38979 - 01/29/2013 10:00 PM - marcandre (Marc-Andre Lafortune)**

- object.c: Improve error for failed implicit conversions [Bug #7539]
- error.c: Adapt rdoc
- test/ruby/test\_object.rb: Test for above

**Revision 38979 - 01/29/2013 10:00 PM - marcandre (Marc-Andre Lafortune)**

- object.c: Improve error for failed implicit conversions [Bug #7539]
- error.c: Adapt rdoc
- test/ruby/test\_object.rb: Test for above

**Revision 3df94d7b - 01/29/2013 10:15 PM - marcandre (Marc-Andre Lafortune)**

- re.c (reg\_operand): Simplify and reuse error handling [Bug #7539]
- test/ruby/test\_regexp.rb: Test for above

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@38980 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 38980 - 01/29/2013 10:15 PM - marcandre (Marc-Andre Lafortune)**

- re.c (reg\_operand): Simplify and reuse error handling [Bug #7539]
- test/ruby/test\_regexp.rb: Test for above

**Revision 38980 - 01/29/2013 10:15 PM - marcandre (Marc-Andre Lafortune)**

- re.c (reg\_operand): Simplify and reuse error handling [Bug #7539]
- test/ruby/test\_regexp.rb: Test for above

**Revision 38980 - 01/29/2013 10:15 PM - marcandre (Marc-Andre Lafortune)**

- re.c (reg\_operand): Simplify and reuse error handling [Bug #7539]
- test/ruby/test\_regexp.rb: Test for above

**Revision 38980 - 01/29/2013 10:15 PM - marcandre (Marc-Andre Lafortune)**

- re.c (reg\_operand): Simplify and reuse error handling [Bug #7539]
- test/ruby/test\_regexp.rb: Test for above

**Revision 38980 - 01/29/2013 10:15 PM - marcandre (Marc-Andre Lafortune)**

- re.c (reg\_operand): Simplify and reuse error handling [Bug #7539]
- test/ruby/test\_regexp.rb: Test for above

**Revision 38980 - 01/29/2013 10:15 PM - marcandre (Marc-Andre Lafortune)**

- re.c (reg\_operand): Simplify and reuse error handling [Bug #7539]
- test/ruby/test\_regexp.rb: Test for above

## History

---

**#1 - 12/10/2012 09:52 PM - kosaki (Motohiro KOSAKI)**

I support "can't append nil to string" error.

**#2 - 12/11/2012 12:44 AM - marcandre (Marc-Andre Lafortune)**

You'll get this error message for every failed conversion; the use the corresponding implicit conversion methods (to\_str, to\_int, ...)

```
" " + 42  
[].concat(1..2)
```

I'm surprised this leads to confusion, but we could change the message to something like:

```
"no implicit conversion of nil into String"
```

**#3 - 12/11/2012 01:44 AM - rosenfeld (Rodrigo Rosenfeld Rosas)**

kosaki (Motohiro KOSAKI) wrote:

I support "can't append nil to string" error.

I'd prefer a more descriptive message like:

"can't append nil to string (object doesn't respond to to\_str)"

This way someone would know that this would work:

```
nil.class.module_eval{alias to_str to_s}; " + nil
```

#### #4 - 12/11/2012 06:23 AM - phluid61 (Matthew Kerwin)

I support adding the 'implicit' word to the error message.

```
"no implicit conversion of nil into String"
```

I remember when I first encountered it, knowing to search for "implicit conversion" would have helped me understand the difference between #to\_s and #to\_str earlier.

[note: it's more than just implementing #to\_str, there are other implications for implicit casting]

#### #5 - 12/11/2012 01:49 PM - duerst (Martin Dürst)

- File patch#7539.txt added

- Target version changed from 1.9.3 to 2.6

rosenfeld (Rodrigo Rosenfeld Rosas) wrote:

kosaki (Motohiro KOSAKI) wrote:

I support "can't append nil to string" error.

I'd prefer a more descriptive message like:

"can't append nil to string (object doesn't respond to to\_str)"

Looking at the implementation, a "can't append" would need some special code. A message such as ""can't convert nil into String (using to\_str)" would be relatively easy to generate. I have attached a patch (not compiled or tested, sorry). This change would show up in many other places that use the same conversion logic, so let's make sure we really want it.

#### #6 - 12/11/2012 02:17 PM - drbrain (Eric Hodel)

I am worried about implying that the user should add to\_str, to\_ary, etc. to fix this problem. Rarely is implementing these methods appropriate since they are used for objects that are intended to behave exactly as a String, Array, etc.

An error message that omitted mention of these methods can help prevent users from accidentally shooting themselves in the foot.

#### #7 - 12/11/2012 11:52 PM - marcandre (Marc-Andre Lafortune)

- Assignee set to marcandre (Marc-Andre Lafortune)

Looking at the implementation, a "can't append" would need some special code

Indeed. It would be redundant anyways, since the error message starts with "in `+":", or whichever method was called last.

I am worried about implying that the user should add to\_str, to\_ary, etc. to fix this problem

Agreed, which is why I proposed making explicit the fact that implicit conversion failed.

So, I'll commit "no implicit conversion of into " unless there's a better wording proposed

**#8 - 01/30/2013 07:01 AM - marcandre (Marc-Andre Lafortune)**

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset [r38979](#).

Chris, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

- 
- object.c: Improve error for failed implicit conversions [Bug [#7539](#)]
  - error.c: Adapt rdoc
  - test/ruby/test\_object.rb: Test for above

**Files**

---

patch#7539.txt	583 Bytes	12/11/2012	duerst (Martin Dürst)
----------------	-----------	------------	-----------------------