Ruby master - Bug #5915
Array#join with explicit nil should not use $,
01/21/2012 03:26 AM - john_firebaugh (John Firebaugh)

| Status:   | Closed                        |
| Priority: | Normal                        |
| Assignee: | naruse (Yui NARUSE)           |
| Target version: | 2.0.0                      |

ruby -v:
ruby 1.9.3p0 (2011-10-30 revision 33570) [x86_64-darwin10.8.0]

Backport:

Description
=begin
The documentation for Array#join says that the default argument is $,. But:

$, = ",
[1,2,3].join(nil) #=> "1,2,3"

Based on the documentation (and the POLS), I would expect "123".
=end

Associated revisions
Revision 61a67bc7 - 07/25/2012 07:13 AM - naruse (Yui NARUSE)
Add test to clarify r36421. [Bug #5915]

Revision 36535 - 07/25/2012 07:13 AM - naruse (Yui NARUSE)
Add test to clarify r36421. [Bug #5915]

Revision 36535 - 07/25/2012 07:13 AM -naruse (Yui NARUSE)
Add test to clarify r36421. [Bug #5915]

Revision 36535 - 07/25/2012 07:13 AM - naruse (Yui NARUSE)
Add test to clarify r36421. [Bug #5915]

Revision 36535 - 07/25/2012 07:13 AM - naruse (Yui NARUSE)
Add test to clarify r36421. [Bug #5915]

Revision 36535 - 07/25/2012 07:13 AM - naruse (Yui NARUSE)
Add test to clarify r36421. [Bug #5915]

Revision 36535 - 07/25/2012 07:13 AM - naruse (Yui NARUSE)
Add test to clarify r36421. [Bug #5915]

History
#1 - 01/21/2012 01:18 PM - naruse (Yui NARUSE)

John Firebaugh wrote:

The documentation for Array#join says that the default argument is $,. But:

$, = ",
[1,2,3].join(nil) #=> "1,2,3"

Based on the documentation (and the POLS), I would expect "123".

nil is not "" , so there is no such documentation.
On Fri, Jan 20, 2012 at 8:19 PM, Yui NARUSE naruse@airemix.jp wrote:

nil is not "", so there is no such documentation.

I don't understand your point.

To quote from array.c:

```c
/*
 call-seq:
 ary.join(sep=nil)    -> str *
 Returns a string created by converting each element of the array to
 a string, separated by +sep+. *
 ["a", "b", "c"].join #=> "abc"
 ["a", "b", "c"].join("") #=> "a-b-c" */
```

Clearly $, is documented as being the default. If that is not the desired behavior, the documentation should be changed. I would suggest:

```c
/*
 call-seq:
 ary.join(sep=nil)    -> str *
 Returns a string created by converting each element of the array to
 a string, separated by +sep+. If +sep+ is nil, $, is used instead. *
 ["a", "b", "c"]join #=> "abc"
 ["a", "b", "c"].join("") #=> "a-b-c" */
```

But I would prefer it to work as currently documented, because that behavior is (slightly) less complex in that it doesn't require explicitly specifying that "If +sep+ is nil, $, is used instead."

#3 - 01/21/2012 01:55 PM - john_firebaugh (John Firebaugh)

$, = nil
[1, 2, 3].join

$, = ",
[1, 2, 3].join(nil)

According to how #join is currently documented, these should produce the same result.

#4 - 01/22/2012 01:47 AM - marcandre (Marc-Andre Lafortune)

I feel that both Yui and John are right.

As Yui points out, the documentation doesn't state explicitly how nil is treated.

On the other hand, I agree with John that the current behaviour is not intuitive and that either the documentation should be clarified or the behaviour should be changed.

1) To explicitly describe the current behavior, the documentation needs to state that if sep is nil, then $, is used and that if $, is nil, then no separator is used. This last part is also needed, because currently Array#join does not call #to_s on sep, so nil is actually a special value.

Or else:
2) The current behavior so that sep == nil means that there will be no separator. Documentation could state this explicitly by saying that "... separated by +sep+ (if non-nil)". Clearly, the shorter documentation reflects the fact that this is simpler. I believe this is what John suggests.

3) The behaviour could also be changed so that Array#join calls #to_s on sep, like it does on its members. Most Ruby functions only call to_str on the arguments, but Array#join calls to_s on the elements of the array, so it's not clear (to me) why it doesn't on sep (maybe Matz has an idea?). In that case, the behavior would match what John and I favor, and the documentation doesn't really need to change or could state that "elements and sep will be converted to strings if need be by calling to_s".

I personally favor (2) or (3), so that:

$, = "foo"
[4,2].join(nil) #=> "4foo2"

Yui, do you prefer the current behavior where the result is "4foo2"?
Marc, thank you for clarification.

Marc-Andre Lafortune wrote:

As Yui points out, the documentation doesn’t state explicitly how nil is treated.

Yes.

On the other hand, I agree with John that the current behaviour is not intuitive and that either the documentation should be clarified or the behaviour should be changed.

1) To explicitly describe the current behavior, the documentation needs to state that if sep is nil, then $, is used and that if $, is nil, then no separator is used. This last part is also needed, because currently Array#join does not call #to_s on sep, so nil is actually a special value.

Yes, there is this option.
People must show the reason against this.

Or else:
2) The current behavior so that sep == nil means that there will be no separator. Documentation could state this explicitly by saying that "... separated by +sep+ (if non-nil)". Clearly, the shorter documentation reflects the fact that this is simpler. I believe this is what John suggests.

3) The behaviour could also be changed so that Array#join calls #to_s on sep, like it does on its members. Most Ruby functions only call to_str on the arguments, but Array#join calls to_s on the elements of the array, so it's not clear (to me) why it doesn't on sep (maybe Matz has an idea?). In that case, the behavior would match what John and I favor, and the documentation doesn't really need to change or could state that "elements and sep will be converted to strings if need be by calling to_s".

I personally favor (2) or (3), so that:

$, = "foo"
[4,2].join(nil) == "42"

Yui, do you prefer the current behavior where the result is "4foo2"?

nil is a special value so that it can occur some special behavior.
The point seems the difference between $,=nil and sep=nil, feel strange.

John Firebaugh wrote:

Array#join actually tries three coercions on elements: #to_str, followed by #to_ary (for nested joins), followed by #to_s.

The separator is coerced with #to_str if it is not nil and not a String. Note that this matters only for an explicitly passed parameter, as assignment to $, is restricted to Strings and nil. I would favor lifting this restriction, as the current coercion logic is applied to $, -- it just happens to be a no-op given the assignment restriction.

Charles Nutter wrote:

matz said this is documentation bug.
I'll fix it.

Added in r36421.
Is there a test for this? If not, perhaps one should be added.

I'm fixing http://jira.codehaus.org/browse/JRUBY-6776 and would like to know that the behavior is being tested.