Ruby trunk - Bug #7964

Writing an ASCII-8BIT String to a StringIO created from a UTF-8 String

02/26/2013 04:32 PM - brixen (Brian Shirai)

Status:	Assigned		
Priority:	Normal		
Assignee:	nobu (Nobuyoshi Nakada)		
Target version:			
ruby -v:	ruby 2.0.0p0 (2013-02-24 revision 39474) [x86_64-darwin10.8.0]	**Backport:**	

Description

=begin
In the following script, an ASCII-8BIT String is written to a StringIO created with a UTF-8 String without error. However, a << b or a + b will raise an exception, as will writing an ASCII-8BIT String to a File with UTF-8 external encoding.

- $ cat file_enc.rb # encoding: utf-8 require 'stringio'

a = "On a very cold morning, it was -8°F."
b = a.dup.force_encoding "ascii-8bit"

io = StringIO.new a
io.write(b)
p io.string.encoding

File.open "data.txt", "w:utf-8" do |f|
f.write a
f.write b
end

- $ ruby2.0 -v file_enc.rb
  ruby 2.0.0p0 (2013-02-24 revision 39474) [x86_64-darwin10.8.0]
  #Encoding:UTF-8
  file_enc.rb:13:in write': "\xC2" from ASCII-8BIT to UTF-8 (Encoding::UndefinedConversionError)
  from file_enc.rb:13:inblock in '
  from file_enc.rb:11:in open'
  from file_enc.rb:11:in'

- $ ruby1.9.3 -v file_enc.rb
  ruby 1.9.3p327 (2012-11-10 revision 37606) [x86_64-darwin10.8.0]
  #Encoding:UTF-8
  file_enc.rb:13:in write': "\xC2" from ASCII-8BIT to UTF-8 (Encoding::UndefinedConversionError)
  from file_enc.rb:13:inblock in '
  from file_enc.rb:11:in open'
  from file_enc.rb:11:in'
  =end

History

#1 - 02/26/2013 04:50 PM - nobu (Nobuyoshi Nakada)

- Description updated

#2 - 02/26/2013 04:56 PM - nobu (Nobuyoshi Nakada)

- Category set to ext

- Status changed from Open to Assigned

- Assignee set to nobu (Nobuyoshi Nakada)

- Target version set to 2.1.0

Currently, StringIO does not support encoding conversion on write, so `io.write(b)' does not raise any exceptions.

#3 - 02/26/2013 06:18 PM - duerst (Martin Dürst)

nobu (Nobuyoshi Nakada) wrote:

> Currently, StringIO does not support encoding conversion on write, so `io.write(b)' does not raise any exceptions.

Should StringIO support encoding conversion? I think it should, because it should work like IO. However, the question is whether the resulting string should always be BINARY (exactly mirroring what happens with real IO), or whether it should have its own encoding (this might allow collecting substrings in different encodings into a string with a single encoding without any explicit conversions).

I think that somebody should open a feature for this, and of course patches would be welcome.

As an aside, I think it would be easier implementing StingIO in Ruby, or is StringIO performance critical?

#4 - 02/26/2013 07:37 PM - naruse (Yui NARUSE)

The examples are not equal.
Correct comparison is

```
StringIO.open a, "w" do |io|
io.write(b)
end

File.open "data.txt", "w" do |io|
io.write b
end
```

So it won't raise error even if StringIO supports external/internal encoding.

duerst (Martin Dürst) wrote:

> nobu (Nobuyoshi Nakada) wrote:
>
> > Currently, StringIO does not support encoding conversion on write, so `io.write(b)' does not raise any exceptions.
>
> Should StringIO support encoding conversion? I think it should, because it should work like IO. However, the question is whether the resulting string should always be BINARY (exactly mirroring what happens with real IO), or whether it should have its own encoding (this might allow collecting substrings in different encodings into a string with a single encoding without any explicit conversions).

Agreed.

> I think that somebody should open a feature for this, and of course patches would be welcome.
>
> As an aside, I think it would be easier implementing StingIO in Ruby, or is StringIO performance critical?

see https://bugs.ruby-lang.org/issues/5677

#5 - 02/27/2013 10:38 AM - brixen (Brian Shirai)

Martin, what do you mean by: "However, the question is whether the resulting string should always be BINARY (exactly mirroring what happens with real IO)..."?

If StringIO is going to fake aliasing #pos across instances that have been #dup'd, it should certainly have the same encoding-related behavior. Cf. http://bugs.ruby-lang.org/issues/7220

#6 - 01/30/2014 06:16 AM - hsbt (Hiroshi SHIBATA)

- Target version changed from 2.1.0 to 2.2.0

#7 - 01/05/2018 09:00 PM - naruse (Yui NARUSE)

- Target version deleted (2.2.0)