The result of UTF-16 encoded string concatenation

12/10/2010 01:59 PM - phasis68 (Heesob Park)

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

Description
=begin
C:\work>irb
irb(main):001:0> a = 'abc'.encode('UTF-16')
=> ﻿abc
irb(main):002:0> b = a + a
=> ﻿abc﻿abc
irb(main):003:0> c = b.encode('UTF-8')
=> "abc﻿abc"
irb(main):004:0> d = b.encode('US-ASCII')
Encoding::UndefinedConversionError: U+FEFF to US-ASCII in conversion from UTF-16 to UTF-8 to US-ASCII
from (irb):4:in encode'
from (irb):4
from c:/usr/bin/irb.bat:19:in'
irb(main):005:0> b << b
=> ﻿abc﻿abc﻿abc﻿abc
irb(main):006:0> b * 3
=> "abc﻿abc﻿abc\uFEFFabc\uFEFFabc\uFEFFabc\uFEFFabc\uFEFFabc\uFEFFabc\uFEFFabc\uFEFFabc\uFEFFabc\uFEFFabc\uFEFFabc\uFEFFabc\uFEFFabc"
irb(main):007:0>

Although I understand this behaviour, is there any possibility of generating only one \uFEFF ?
=end

History
#1 - 12/10/2010 02:53 PM - naruse (Yui NARUSE)
- Status changed from Open to Assigned
- Assignee set to naruse (Yui NARUSE)

=begin
Strings encoded in UTF-16 don't support concatenation.
Use UTF-16BE or UTF-16LE for processing.

I'm considering to warn concatenation of strings encoded in dummy encoding.
=end

#2 - 12/10/2010 06:14 PM - duerst (Martin Dürst)

=begin
We should try to get a better overall idea of what "UTF-16" and so on are for. I asked some questions at the very end of [ruby-core:33461]. Yui, can you try to give answers? I hope this will help having a general discussion of the issues involved.

Regards, Martin.

On 2010/12/10 14:53, Yui NARUSE wrote:

Issue #4145 has been updated by Yui NARUSE.

Status changed from Open to Assigned
Assigned to set to Yui NARUSE
Strings encoded in UTF-16 don't support concatenation. Use UTF-16BE or UTF-16LE for processing.

I'm considering to warn concatenation of strings encoded in dummy encoding.

http://redmine.ruby-lang.org/issues/show/4145

---

#-# Martin J. Dürst, Professor, Aoyama Gakuin University
{-# http://www.sw.it.aoyama.ac.jp mailto:duerst@it.aoyama.ac.jp
=end

#3 - 12/10/2010 06:15 PM - duerst (Martin Dürst)

=begin
We should try to get a better overall idea of what "UTF-16" and so on are for. I asked some questions at the very end of [ruby-core:33461]. Yui, can you try to give answers? I hope this will help having a general discussion of the issues involved.

Regards, Martin.

On 2010/12/10 14:53, Yui NARUSE wrote:

Issue #4145 has been updated by Yui NARUSE.

Status changed from Open to Assigned
Assigned to set to Yui NARUSE

Strings encoded in UTF-16 don't support concatenation. Use UTF-16BE or UTF-16LE for processing.

I'm considering to warn concatenation of strings encoded in dummy encoding.

http://redmine.ruby-lang.org/issues/show/4145

---

#-# Martin J. Dürst, Professor, Aoyama Gakuin University
{-# http://www.sw.it.aoyama.ac.jp mailto:duerst@it.aoyama.ac.jp
=end

#4 - 12/11/2010 04:09 AM - naruse (Yui NARUSE)

=begin
(2010/12/10 18:14), "Martin J. Dürst" wrote:

We should try to get a better overall idea of what "UTF-16" and so on are for. I asked some questions at the very end of [ruby-core:33461]. Yui, can you try to give answers? I hope this will help having a general discussion of the issues involved.

Current implementation is what I thought to be.

My main questions here are:
A) Which one of the above is the current Ruby implementation effort (the above patch and a few related ones) targetting?

This is, 2b) XML strictly requires a BOM. Because the spec (2a) collides the real (2c).

B) How complete is that implementation (thought to be)?

09/13/2021
Current one is completed one.

C) What about other implementation needs?

Nothing, in current situation.

D) What can we do to make sure users have at least a chance of understanding what "UTF-16" in Ruby is good for?

This is open problem, but so I implement it and am seeing user's reactions.

--
NARUSE, Yui naruse@airemix.jp

=end

#5 - 12/26/2010 03:21 PM - naruse (Yui NARUSE)
- Status changed from Assigned to Closed

=start
=end