

Ruby master - Bug #7201

Setting default_external affects STDIN encoding but default_internal does not

10/21/2012 02:29 PM - brixen (Brian Shirai)

Status: Rejected	
Priority: Normal	
Assignee: naruse (Yui NARUSE)	
Target version: 2.0.0	
ruby -v: ruby 1.9.3p286 (2012-10-12 revision 37165) [x86_64-darwin10.8.0]	Backport:
Description	
Changing Encoding.default_external changes STDIN.external_encoding, but changing Encoding.default_internal does not change STDIN.internal_encoding.	
STDOUT and STDERR internal/external encodings are not changed in either case and are always nil.	
Is this a bug? See the following IRB transcript:	
<pre>\$ irb 1.9.3p286 :001 > Encoding.default_external => #Encoding:UTF-8 1.9.3p286 :002 > Encoding.default_internal => nil 1.9.3p286 :003 > STDIN.external_encoding => #Encoding:UTF-8 1.9.3p286 :004 > STDIN.internal_encoding => nil 1.9.3p286 :005 > Encoding.default_external = "euc-jp" => "euc-jp" 1.9.3p286 :006 > STDIN.external_encoding => #Encoding:EUC-JP 1.9.3p286 :007 > STDIN.internal_encoding => nil 1.9.3p286 :008 > Encoding.default_internal = "iso-8859-1" => "iso-8859-1" 1.9.3p286 :009 > STDIN.internal_encoding => nil</pre>	
Thanks, Brian	

History

#1 - 11/05/2012 09:58 PM - mame (Yusuke Endoh)

- Status changed from Open to Assigned
- Assignee set to naruse (Yui NARUSE)
- Target version set to 2.0.0

Naruse-san, could you handle this?

--

Yusuke Endoh mame@tsg.ne.jp

#2 - 11/09/2012 05:01 PM - naruse (Yui NARUSE)

- Status changed from Assigned to Rejected

This is not a bug in 1.9.3 and 2.0.0 while I feel this behavior is not so good.
I want to change this but it will be big change, therefore I keep compatibility in near future.

#3 - 11/10/2012 07:58 AM - brixen (Brian Shirai)

Can someone please explain how the inconsistency with how the rest of IO instances would behave with transcoding is not a bug?

Thanks,
Brian

#4 - 11/11/2012 12:23 AM - duerst (Martin Dürst)

Hello Brian,

I'm not sure what the reason was for the current state, but I can easily imagine a situation where stdin/stdout are the console and therefore in one encoding, whereas the data a script is working on is all in another encoding.

Regards, Martin.

#5 - 11/12/2012 02:29 PM - naruse (Yui NARUSE)

brixen (Brian Ford) wrote:

Can someone please explain how the inconsistency with how the rest of IO instances would behave with transcoding is not a bug?

This is because IO object's internal property are set when it is created.
In this case, STDIN's internal property is not changed when default_external and default_internal are set.

And in this situation, STDIN.external_encoding returns current Encoding.default_external, so it looks as if Encoding.default_external changes STDIN.

Following are detail

= IO's internal property

An IO object has two internal properties, extenc (external encoding) and intenc (internal encoding).

When extenc and intenc are explicitly given like open("foo.txt", "r:UTF-8:ISO-8859-1"), extenc is UTF-8 and intenc is ISO-8859-1

When extenc and intenc are not given like open("foo.txt", "r") or STDIN without -E/-U, extenc is nil and intenc is nil.

= IO#external_encoding

If extenc is not nil, returns extenc.
If extenc is nil, returns current Encoding.default_external.

This method is to know what encoding is set on io.read.
(this had to be always return extenc...)

= IO#internal_encoding

Returns intenc.

= Conclusion

Current inconsistency is derived from IO objects' internal state and settings for conversion.
The change will need add more internal property and breaking IO#external_encoding.
I couldn't design better one yet.