Ruby master - Bug #7742
System encoding (Windows-1258) is not recognized by Ruby to convert back to UTF-8

01/26/2013 03:33 PM - Mars (Hong Ha Dang)

Status: Open
Priority: Normal
Assignee: duerst (Martin Dürst)
Target version: ruby -v: 1.9.3

Description
I installed Railsinstaller in win8. After intall complete the screen set to configuration Railsinstaller on cmd (step 2). I give user name: DHH Mars and email: dhhma...@gmail.com. It ran and have following massage:

C:/RailsInstaller/scripts/config_check.rb:64:in 'exist?': code converter not found
Encoding::ConverterNotFoundError from C:/RailsInstaller/scripts/config_check.rb:64:in 'main'
C:\Sites>

Related issues:
Blocked by Ruby master - Bug #6351: transcode table generator does not support... Assigned

History
#1 - 01/26/2013 04:03 PM - duerst (Martin Dürst)
Mars (Hong Ha Dang) wrote:

C:/RailsInstaller/scripts/config_check.rb:64:in 'exist?': code converter not found

Yes, windows-1258 (for Vietnamese) is currently not supported. The reason for this is because conversion from windows-1258 to UTF-8 should produce output in Unicode Normalization Form C. As an example, the sequence 0xE3 0xEC (LATIN SMALL LETTER A WITH BREVE followed by COMBINING ACCUTE ACCENT) should not be converted to the sequence U+0103 U+0301, but to the single character U+1EAF (LATIN SMALL LETTER A WITH BREVE AND ACCUTE).

This means that this bug depends on bug #6351. Unfortunately, I don't have time now to work on that bug; this will have to wait for March, sorry.

#2 - 01/26/2013 04:05 PM - duerst (Martin Dürst)
- Assignee set to duerst (Martin Dürst)
- Target version set to 2.6

#3 - 01/08/2014 09:11 PM - thegcat (Felix Schäfer)

begin
We (((((())))) are also in need of Windows-1258 to UTF-8 conversion, is there anything we can do to help?
=end

#4 - 01/09/2014 04:26 PM - duerst (Martin Dürst)

thegcat (Felix Schäfer) wrote:

begin
We (((((())))) are also in need of Windows-1258 to UTF-8 conversion, is there anything we can do to help?
=end

As explained above, the problem is with normalization. If you are okay with a version that just does one-to-one conversion, then that can be produced rather quickly (maybe even over the weekend). But most Vietnamese content, e.g. on the Web, is normalized (NFC), and I guess you'd want to have that, too. But then you also have to be careful with respect to round-tripping, because windows-1258->UTF-8 will be .encode(‘UTF-8’).
'windows-1258').to_nfc or so, but backwards conversion would need special code because neither NFC nor NFD can directly be converted to windows-1258.

A slightly more elaborate version would do one-to-one conversion from windows-1258 to UTF-8, but would convert that kind of data as well as data in NFC back to windows-1258 (but not arbitrarily non-normalized data) back to windows-1258. Such a converter might be relatively easy to produce, or it might be more difficult; I can't say which off the top of my head.

So if you use a normalization library after conversion, that might work out, but it would be somewhat of a special case. Also, when we later introduce a different (more normalizing) converter, that may be seen as a non-backwards-compatible change.

One solution to backwards-compatibility would be to use different encoding labels to differentiate versions of conversion. But this has the problem that in the current state of affairs, it introduces additional "encodings" that are not really different, but just variants produced by different conversions.

That's the problem e.g. with the current UTF8-MAC, and I don't want to create more of these.

A more long-term solution would be to introduce a difference between encodings and conversions, where e.g. one could use .encode('windows-1258--non-normalized', 'utf-8') or so to indicate a non-normalized version of conversion. But that would need some more general discussion among the Ruby experts in this field.

So Felix, if you tell me what you need, and we can make sure that it doesn't affect later backwards-compatibility, I might be able to work on something.

#5 - 01/09/2014 07:15 PM - phasis68 (Heesob Park)

As I know, VISCII(Vietnamese Standard Code for Information Interchange) can round trip UTF-8. So the implementation of the converter between VISCII and UTF-8 might be easy.

I am not sure if a converter between Windows-1258 and VISCII is possible, Windows-1258 can be supported via VISCII.

Windows-1258 <-> VISCII <-> UTF-8

Anyway, it would be nice if ruby supports VISCII encoding.

#6 - 01/09/2014 08:33 PM - duerst (Martin Dürst)

phasis68 (Heesob Park) wrote:

As I know, VISCII(Vietnamese Standard Code for Information Interchange) can round trip UTF-8. So the implementation of the converter between VISCII and UTF-8 might be easy.

Yes, it should be easy. Can you open a separate ticket? I'll give it a try over the weekend.

I am not sure if a converter between Windows-1258 and VISCII is possible, Windows-1258 can be supported via VISCII.

Conversion between Windows-1258 and VISCII is actually as difficult as the conversion between Windows-1258 and NFC-normalized UTF-8, which is the most difficult variant as I have explained above.

#7 - 12/25/2017 06:15 PM - naruse (Yui NARUSE)

- Target version deleted (2.6)