

Ruby master - Bug #13549

MinGW / Windows encoding - Two issues

05/08/2017 06:55 PM - MSP-Greg (Greg L)

Status: Closed	
Priority: Normal	
Assignee:	
Target version:	
ruby -v: ruby 2.5.0dev (2017-05-08 trunk 58610) [x64-mingw32]	Backport: 2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN

Description

Issue #1

The documentation for [Encoding.default_internal=](#) states:

"The locale encoding (`__ENCODING__`), not `default_internal`, is used as the encoding of created strings."

Below is code and the console output for a MinGW build. Whether a variable is assigned to a string, or a string directly, it appears that both are encoded UTF-8, regardless of the locale encoding.

So, something is amiss. Is it --

1. The documentation mistaken
2. The behavior is specific to *nix builds
3. The MinGW build is behaving incorrectly

```
txt = 'ABCDEF_äÖü'
puts "filesystem   #{Encoding.find('filesystem')} " \
     "\nlocale     #{Encoding.find('locale')} " \
     "\nexternal    #{Encoding.default_external} " \
     "\ninternal     #{Encoding.default_internal} " \
     "\ntxt         #{txt.encoding.to_s} " \
     "\n'ABCDEF_äÖü'  #{'ABCDEF_äÖü'.encoding.to_s}"
```

Console out with default encoding

```
filesystem   Windows-1252
locale       IBM437
external     IBM437
internal
txt          UTF-8
'ABCDEF_äÖü' UTF-8
```

Console out with locale set to 1252 with chcp

```
filesystem   Windows-1252
locale       Windows-1252
external     Windows-1252
internal
txt          UTF-8
'ABCDEF_äÖü' UTF-8
```

Issue #2

In the issue [Set Encoding.default_external to UTF-8 on Windows #13488](#), Lars Kanis proposed changing Ruby default encodings on Windows to UTF-8. Discussion showed that, at present, this would be an issue for many users.

In that thread, Nobu posted console output that showed `default_external` matching `filesystem`.

```
C:\Users\nobu\work\ruby\trunk\x64-mswin32_140>.\bin\ruby -e "p Encoding.default_external, Encoding.find('filesystem')"
```

#<Encoding:Windows-31J>

#<Encoding:Windows-31J>

In recent MinGW builds, I've had 8 failures and 1 error. This weekend I spent a little time patching around three failures, two of which involved encoding. The patches are dependent on the cause/fix for Issue #1, but also seem to work best when locale and default_external encodings are set equal to filesystem.

As noted above, my Windows system (standard American English Win7) has filesystem encoding of Windows-1252, with locale and default_external are IBM437. Why, I don't know.

Given that Nobu showed filesystem equal to default_external, would it be possible to change 'Windows' ruby so that, by default, locale and default_external are set equal to filesystem?

Not being a c type, I cannot create a patch/PR, etc. Lastly, moving this post between my code editor and 'Visual Studio Code' had some encoding issues. Or, yes, Windows does still have encoding issues...

History

#1 - 05/09/2017 08:58 AM - duerst (Martin Dürst)

Please post separate issues separately. If they are related, please link them with "Related issues".

#2 - 07/17/2017 12:04 AM - MSP-Greg (Greg L)

Since I posted this, nobu (thank you) authored a few commits that improved the windows encoding issues (not necessarily related to this issue). Since then I do not recall many encoding related failures, and a few patches I have for such I've disabled.

So, I think nobu's commits solved most of the problems, although I posted an issue related to the fact that File.exist?(fn) was true, but ruby #{fn} did not work. I'll have to check that.

Anyway, over in windows world, we're looking at merging some of my work (patches, custom MinGW packages, testing) back into RubyInstaller2, and I recently ran builds/tests on ruby_2_4 and 2.4.1. I believe some of the encoding failures appeared. Hence, I don't know if nobu's commits were backported or not. That might be helpful.

Otherwise, please close, and thanks for all of your work.

#3 - 10/18/2019 12:29 AM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Closed