Ruby - Bug #9715

ENV data yield ASCII-8BIT encoded strings under Windows with unicode username

04/08/2014 12:08 PM - thomthom (Thomas Thomassen)

Status:	Closed		
Priority:	Normal		
Assignee:	windows		
Target version:			
ruby -v:	ruby 2.2.0dev (2014-04-07 trunk 45530) [i386-mswin32_100]	**Backport:**	2.0.0: UNKNOWN, 2.1: UNKNOWN

Description

My testing scenario:
English Windows, Unicode username: □□□

Home directory: C:\Users\□□□\

The values returned from ENV have different encoding depending on their content. It appear to be OEM encoding label to most value, except when they contain characters not included in the OEM codepage. When they are not, for instance ENV['HOME'] when the username is "□□□" will have ASCII-8BIT.

(I find the "ASCII-8BIT" name for an encoding confusing, as ASCII is 7bit - byte range 0-127)
But it appear that "ASCII-8BIT" is also aliased as "binary"? So Ruby is here returning a binary string when ENV contain byte characters not included in the OEM code page?

Reading the docs for Encoding:

> Returns default internal encoding. Strings will be transcoded to the default internal encoding in the following places if the default internal encoding is not nil:
> ...
> ::default_internal is initialized by the source file's internal_encoding or -E option.

This includes ENV - but, even when I run ruby with the -E flag the ENV encoding doesn't change. It's still using the OEM code page - or ASCII-8BIT.
However, regardless of having set -E or not, ENV do appear to return UTF-8 bytes in the strings that contain the Unicode username.

This is one of several areas where I have found -E to have no effect on Ruby's string handling. I understand that some of Ruby's file handling is for backwards compatibility reasons, but I'm finding it difficult to set up a system which can properly handle Unicode files under Windows. Is this deliberate due to backwards compatibility decisions? Or have I simply not found the correct configuration flags for it? To me it appear bugged - inconsistent with what the documentation says. But please enlighten me if I am incorrect. My ideal situation would be for all strings to default to UTF-8.

Examples:

```
C:\ruby-220\usr\bin>ruby -E UTF-8:UTF-8 -e "p ENV['ProgramFiles'].encoding"
#<Encoding:CP850>

C:\ruby-220\usr\bin>ruby -E UTF-8:UTF-8 -e "p ENV['ProgramFiles'].bytes"
[67, 58, 92, 80, 114, 111, 103, 114, 97, 109, 32, 70, 105, 108, 101, 115, 32, 40, 120, 56, 54, 41]

C:\ruby-220\usr\bin>ruby -e "p ENV['HOME']"
"C:/Users/\xE3\x81\xA6\xE3\x81\x99\xE3\x81\xA8"

C:\ruby-220\usr\bin>ruby -e "p ENV['HOME'].encoding"
#<Encoding:ASCII-8BIT>

C:\ruby-220\usr\bin>ruby -e "p ENV['HOME'].bytes"
[67, 58, 47, 85, 115, 101, 114, 115, 47, 227, 129, 166, 227, 129, 153, 227, 129, 168]

C:\ruby-220\usr\bin>ruby -e "p __ENCODING__"
#<Encoding:CP850>
```

```
C:\ruby-220\usr\bin>ruby -e "p Encoding.default_internal"
nil

C:\ruby-220\usr\bin>ruby -e "p Encoding.default_external"
#<Encoding:CP850>

C:\ruby-220\usr\bin>ruby -e "p Encoding.find('filesystem')"
#<Encoding:Windows-1252>

C:\ruby-220\usr\bin>ruby -E UTF-8:UTF-8 -e "p ENV['HOME'].encoding"
#<Encoding:ASCII-8BIT>

C:\ruby-220\usr\bin>ruby -E UTF-8:UTF-8 -e "p ENV['HOME'].bytes"
[67, 58, 47, 85, 115, 101, 114, 115, 47, 227, 129, 166, 227, 129, 153, 227, 129, 168]
```

Related issues:

Related to Ruby - Feature #12650: Use UTF-8 encoding for ENV on Windows	**Closed**

History

#1 - 12/04/2015 02:09 PM - thomthom (Thomas Thomassen)

Trying to bump this. Even just some feedback on what an acceptable change would be and I could try to produce a patch.

#2 - 12/04/2015 04:27 PM - davispuh (Dāvis Mosāns)

looks like same as bug #8822 and seems it's still wrong even with latest Ruby 2.2

#3 - 12/04/2015 05:37 PM - thomthom (Thomas Thomassen)

Dāvis Mosāns wrote:

> looks like same as bug #8822 and seems it's still wrong even with latest Ruby 2.2

Yes, I just ran my own set of tests on the latest 2.2 release and I'm getting #Encoding:IBM437 for ENV entries despite internal and default encoding set to UTF-8.

#4 - 12/05/2015 03:23 AM - spatulasnout (B Kelly)

Agreed: although Ruby 2.2 appears to be trying to use locale to determine the encoding for environment vars, it's not producing reasonable results on Windows. (E.g. w32_getenv() in hash.c, and also putenv() in hash.c)

Its current behavior seems to mangle UTF-8 data in environment vars so badly that the data can't be reconstructed from within ruby. (As such, simply passing through ASCII-8BIT data untouched would be preferable to the current behavior, as one could at least then force_encoding from within ruby.)

Regards,

Bill

#5 - 06/22/2016 03:42 PM - Iristyle (Ethan Brown)

I agree that ENV corruption continues to be a problem in Ruby. The expectation is that regardless of current locale, I should get UTF-8 strings when using ENV on Windows. Here is a simple reproduction of the failure on 2.3.0:

```
C:\Users\Administrator> $env:unicode = 'task┌┘╢'
C:\Users\Administrator> dir Env:\unicode

Name                           Value
----                           -----
unicode                        task┌┘╢


C:\Users\Administrator> ruby --version
ruby 2.3.0p0 (2015-12-25 revision 53290) [x64-mingw32]
C:\Users\Administrator> chcp
Active code page: 437

C:\Users\Administrator> irb
irb(main):001:0> RUBY_VERSION
=> "2.3.0"
irb(main):002:0> Encoding.default_internal
=> nil
irb(main):003:0> Encoding.default_external
```

```
=> #<Encoding:IBM437>
irb(main):004:0> str = ENV['unicode']
=> "task???"
irb(main):005:0> str.encoding
=> #<Encoding:IBM437>
```

Again, when I access ENV on Windows, I should receive a UTF-8 string with the correct data. The expected string in this case is:

```
irb(main):036:0> str2 = "task\u16A0\u16C7\u16BB"
=> "task\u16A0\u16C7\u16BB"
irb(main):037:0> str2.encoding
=> #<Encoding:UTF-8>
```

Note that some browsers, like Chrome on OSX, may fail to render the Rune characters correctly, but if you copy into a proper editor or use another browser you should see the characters fine.

#6 - 06/23/2016 11:38 PM - davispuh (Dāvis Mosāns)

FYI, I've been working on fixing several (including this) Ruby's Unicode issues on Windows and I hope to submit my patches within next 2 weeks, maybe sooner.

#7 - 09/12/2016 06:38 AM - naruse (Yui NARUSE)

- Related to Feature #12650: Use UTF-8 encoding for ENV on Windows added

#8 - 02/26/2021 10:09 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Closed

As of Ruby 3.0, ENV values are now UTF-8 encoded on Windows.