Ruby master - Feature #10002

String swapcase

07/01/2014 02:55 PM - randy15 (Andreas Runk)

Status: Closed
Priority: Normal
Assignee:
Target version:

Description
Hi, the ruby version 2.1.2 has a problem with the .swapcase function and german letters.
E.g. "ä".swapcase does return "ä" but should be "Ä".

Related issues:
Related to Ruby master - Feature #10085: Add non-ASCII case conversion to String... Closed

History

#1 - 07/01/2014 06:21 PM - nobu (Nobuyoshi Nakada)
- Tracker changed from Bug to Feature

#2 - 07/02/2014 12:09 AM - matz (Yukihiro Matsumoto)
The current implementation of case conversion methods in String class only understands ASCII characters.
We'd like to enhance it when possible. But we have to know how each character should be converted.
For example, how should we convert "ß" (eszett)?

Matz.

#3 - 07/02/2014 01:11 AM - naruse (Yui NARUSE)
At this time, ffi-icu or twitter-text-rb is useful.

#4 - 07/02/2014 01:15 AM - davispuh (Dāvis Mosāns)
It have been already figured out by Unicode Standard, so just have to implement it. Look at Default Case Algorithms in section 3.13 and Case Mappings in section 5.18. Mappings can be viewed in SpecialCasing.txt (and UnicodeData.txt) also CaseFolding.txt could be useful.

From there "ß" (LATIN SMALL LETTER SHARP S) in uppercase would be "SS" (LATIN CAPITAL LETTER S) and it's user's responsibility to know that generally they are not reversible.
Also useful to read Character Properties, Case Mappings & Names FAQ

#5 - 07/02/2014 01:51 AM - zzak (Zachary Scott)
We should delegate to @emboss everytime we need to convert ß...

#6 - 07/02/2014 09:00 AM - shyouhei (Shyouhei Urabe)
We are talking about swapcase, not folding. The "generally they are not reversible" you say is the difficulty we are facing here. Also as you cited CaseFolding.txt, you should have been aware of type T folding, which is impossible without locale information.

If you think you can implement it, please show us.

Dāvis Mosāns wrote:

It have been already figured out by Unicode Standard, so just have to implement it. Look at Default Case Algorithms in section 3.13 and Case Mappings in section 5.18. Mappings can be viewed in SpecialCasing.txt (and UnicodeData.txt) also CaseFolding.txt could be useful.

From there "ß" (LATIN SMALL LETTER SHARP S) in uppercase would be "SS" (LATIN CAPITAL LETTER S) and it's user's responsibility to know that generally they are not reversible.
Also useful to read Character Properties, Case Mappings & Names FAQ

#7 - 07/23/2014 11:07 AM - duerst (Martin Dürst)
- Related to Feature #10085: Add non-ASCII case conversion to String#upcase/downcase/swapcase/capitalize added
This has actually been implemented by Feature #10085, so it can be closed.