### How to deal with capitalizing Georgian in Unicode 11.0.0

06/10/2018 01:03 AM - duerst (Martin Dürst)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>duerst (Martin Dürst)</td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

This is a request for feedback. In particular if you are from Georgia (the country, not the US state), or if you know somebody (who knows somebody,...) from Georgia, feedback on this issue is very much appreciated. If I don't get any feedback, I'll proceed as explained below.

Unicode 11.0.0 introduces an upper-case version of present-day Georgian letters called Mtavruli (the lower case letters are called Mkhedruli). Mtavruli letters are only used to emphasize whole words; there is no initial-letter capitalization in Georgian. Therefore, the Mkhedruli letters do not have Mtavruli letters as their titlecase, but are explicitly mapped to themselves. This means that in Ruby, `mkhedruli.capitalize` would be a no-op although `mkhedruli.upcase` would convert to Mtavruli letters.

**Additional pointers:**

- [http://www.unicode.org/versions/Unicode11.0.0/#Migration](http://www.unicode.org/versions/Unicode11.0.0/#Migration)
- [http://www.unicode.org/versions/Unicode11.0.0/ch07.pdf](http://www.unicode.org/versions/Unicode11.0.0/ch07.pdf)

(Section 7.7, Georgian, pp. 320-321)

**Related issues:**

| Blocks Ruby master - Feature #14802: Update Unicode data to Unicode Version 1... | Closed |

**Revision 3628eae2 - 12/09/2018 11:14 PM - duerst (Martin Dürst)**

implement special behavior for Georgian for `String#capitalize`

The modern Georgian script is special in that it has an ‘uppercase’ variant called MTAVRULI which can be used for emphasis of whole words, for screamy headlines, and so on. However, in contrast to all other bicameral scripts, there is no usage of capitalizing the first letter in a word or a sentence. Words with mixed capitalization are not used at all.

We therefore implement special behavior for `String#capitalize`. Formally, we define `String#capitalize` as first applying `String#downcase` for the whole string, then using `titlecase` on the first letter. Because Georgian defines `titlecase` as the identity function both for MTAVRULI (‘uppercase’) and Mkhedruli (lowercase), this results in `String#capitalize` being equivalent to `String#downcase` for Georgian. This avoids undesirable mixed case.

- `enc/unicode.c`: Actual implementation
- `string.c`: Add mention of this special case for documentation
- `test/ruby/enc/test_case_mapping.rb`: Add two tests, a general one that uses `String#capitalize` on some (including nonsensical) combinations of MTAVRULI and Mkhedruli, and a canary test to detect the potential assignment of characters to the currently open slots (holes) at U+1CBB and U+1CBC.
- `test/ruby/enc/test_case_comprehensive.rb`: Tweak generation of expectation data.

Together with revision 3628eae2, this closes issue #14839.

**Revision 66300 - 12/09/2018 11:14 PM - duerst (Martin Dürst)**

implement special behavior for Georgian for `String#capitalize`
The modern Georgian script is special in that it has an 'uppercase' variant called MTAVRULI which can be used for emphasis of whole words, for screamy headlines, and so on. However, in contrast to all other bicameral scripts, there is no usage of capitalizing the first letter in a word or a sentence. Words with mixed capitalization are not used at all.

We therefore implement special behavior for String#capitalize. Formally, we define String#capitalize as first applying String#downcase for the whole string, then using titlecase on the first letter. Because Georgian defines titlecase as the identity function both for MTAVRULI ('uppercase') and Mkhedruli (lowercase), this results in String#capitalize being equivalent to String#downcase for Georgian. This avoids undesirable mixed case.

- enc/unicode.c: Actual implementation
- string.c: Add mention of this special case for documentation
- test/ruby/enc/test_case_mapping.rb: Add two tests, a general one that uses String#capitalize on some (including nonsensical) combinations of MTAVRULI and Mkhedruli, and a canary test to detect the potential assignment of characters to the currently open slots (holes) at U+1CBB and U+1CBC.
- test/ruby/enc/test_case_comprehensive.rb: Tweak generation of expectation data.

Together with r65933, this closes issue #14839.

Revision 66300 - 12/09/2018 11:14 PM - duerst (Martin Dürst)
implement special behavior for Georgian for String#capitalize

The modern Georgian script is special in that it has an 'uppercase' variant called MTAVRULI which can be used for emphasis of whole words, for screamy headlines, and so on. However, in contrast to all other bicameral scripts, there is no usage of capitalizing the first letter in a word or a sentence. Words with mixed capitalization are not used at all.

We therefore implement special behavior for String#capitalize. Formally, we define String#capitalize as first applying String#downcase for the whole string, then using titlecase on the first letter. Because Georgian defines titlecase as the identity function both for MTAVRULI ('uppercase') and Mkhedruli (lowercase), this results in String#capitalize being equivalent to String#downcase for Georgian. This avoids undesirable mixed case.

- enc/unicode.c: Actual implementation
- string.c: Add mention of this special case for documentation
- test/ruby/enc/test_case_mapping.rb: Add two tests, a general one that uses String#capitalize on some (including nonsensical) combinations of MTAVRULI and Mkhedruli, and a canary test to detect the potential assignment of characters to the currently open slots (holes) at U+1CBB and U+1CBC.
- test/ruby/enc/test_case_comprehensive.rb: Tweak generation of expectation data.

Together with r65933, this closes issue #14839.

History
#1 - 06/10/2018 01:04 AM - duerst (Martin Dürst)
- Blocks Feature #14802: Update Unicode data to Unicode Version 11.0.0 added

#2 - 06/10/2018 01:00 PM - shevegen (Robert A. Heiler)
In other words, we are looking for ruby hackers from Georgia!
Since there are ruby users in nearby Russia and Turkey (Türkiye), this should not be an impossible task.

#3 - 10/01/2018 08:53 AM - duerst (Martin Dürst)

Some notes summarizing some discussions on Unicode-related lists and my current conclusions from these discussions:

- One problem is that fonts supporting MTAVRULI (using upper case to make it easier for everybody) are not yet available. This is a problem that should be solved in a couple of years. It is a problem for applications that use ALL CAPS programmatically converted from something else. This is a problem that should be solved in a couple of years.

- MTAVRULI may not be used in the same contexts as Upper Case in other scripts. One very clear case is that MTAVRULI is only used for ALL CAPS. But this is covered by Unicode data, which means that capitalize will be a no-op. The main area I can see where this can create problems is "Convention over Configuration" situations where all of lowercase, Uppercase, and ALLCAPS are used. If only lowercase and Uppercase are used, Georgian can be treated as an uncameral (only one case) script, similar to e.g. Hiragana. If only lowercase and ALLCAPS are used, then Georgian can be treated as a bicameral (two cases) script.

- Some people (including at some point, myself) have suggested that some of the problems above (e.g. missing fonts) may be addressed by options selecting the pre-version-11-behavior or the new behavior. But making the old behavior default would mean that the new (assumed to be better) behavior would need an option that would rarely be tested in practice but would have to be kept going into the future. Keeping the new behavior as default would mean that old systems would have to be patched, in which case it's better to patch the fonts. So my current thinking is that such an option is overkill.

#4 - 10/02/2018 07:28 AM - duerst (Martin Dürst)

- Tracker changed from Misc to Feature

Changed from Misc to Feature. The Feature would be to add some option(s) to relevant methods such as String#upcase. The baseline (Feature rejected) is that there is no need for options.

String#downcase is unproblematic. String#swapcase is questionable anyway, but assuming there are only monocase (all lower or ALL UPPER) strings in Georgian, it would work fine. It would only produce (non-acceptable) mixed case when starting from (supposedly non-existing) mixed case.

I just noticed that String.capitalize is actually more difficult than I thought. It is a no-op when applied to lowercase, but it will produce mixed case when applied to all uppercase text.

#5 - 10/03/2018 05:45 AM - duerst (Martin Dürst)

duerst (Martin Dürst) wrote:

I just noticed that String.capitalize is actually more difficult than I thought. It is a no-op when applied to lowercase, but it will produce mixed case when applied to all uppercase text.

On the Unicode mailing list, I got the following ideas:

- Provide an option to keep non-start characters (from Markus Scherer, this is available in ICU, see https://www.unicode.org/mail-arch/unicode-ml/y2018-m10/0010.html)
- Formally (re)define str.capitalize as str.downcase.capitalize (from Ken Whistler, see https://www.unicode.org/mail-arch/unicode-ml/y2018-m10/0013.html). This should not change anything for other scripts, but for Georgian, capitalize and downcase would be the same, and capitalize would not produce mixed-case words.

I'm currently leaning towards the second proposal. It looks like this may make the operation a lot slower, but I think it's easy to avoid a major slowdown.

#6 - 10/04/2018 10:34 AM - spixi (Marius Spix)

The current implementation of String.capitalize is not just a problem in Georgian, but also in other languages like Dutch. Words beginning with „ij“ must be titlecased with a leading „IJ“. E.g. „IJsbeer“ (polar bear). This should be also considered when thinking about redesigning the case mapping code.

#7 - 10/08/2018 10:25 AM - duerst (Martin Dürst)

spixi (Marius Spix) wrote:

The current implementation of String.capitalize is not just a problem in Georgian, but also in other languages like Dutch. Words beginning with „ij“ must be titlecased with a leading „IJ“. E.g. „IJsbeer“ (polar bear). This should be also considered when thinking about redesigning the case mapping code.

Thanks for this information. The problem with this is that it is language-specific, i.e. it doesn't apply to all words starting with "ij" in all languages. Also, there's a character, 'i', that correctly upcases to 'U'. Unfortunately, it's not very much used in Dutch text.
Link to (request for) feedback on this issue from Rails: https://groups.google.com/forum/#!topic/rubyonrails-core/fZUk1qXRT5k.

Hey all, I'm from Georgia so I hope I can help. I'm also Ruby dev and I heard about this issue from Akira Matsuda's post on Georgian Ruby Community Facebook page. So as I understood you are trying to handle situation when someone calls string manipulation methods on Georgian. Georgian as mentioned, is a single case alphabet so, converting mkhedruli letters to mtavruli ones upon calling .upcase is going against the OOP intuition. upcase, capitalize, lowercase and other methods like these shouldnt do anything to Georgian Unicode because these methods simply do not apply. Let me know if I missed the point and if the problem is something else.

This is a request for feedback. In particular if you are from Georgia (the country, not the US state), or if you know somebody (who knows somebody,...) from Georgia, feedback on this issue is very much appreciated. If I don't get any feedback, I'll precede as explained below.

Unicode 11.0.0 introduces an upper-case version of present-day Georgian letters called Mtavruli (the lower case letters are called Mkhedruli). Mtavruli letters are only used to emphasize whole words; there is no initial-letter capitalization in Georgian. Therefore, the Mkhedruli letters do not have Mtavruli letters as their titlecase, but are explicitly mapped to themselves. This means that in Ruby, mkhedruli.capitalize would be a no-op although mkhedruli.upcase would convert to Mtavruli letters.

Additional pointers:
http://www.unicode.org/versions/Unicode11.0.0/#Migration
http://www.unicode.org/versions/Unicode11.0.0/ch07.pdf (Section 7.7, Georgian, pp. 320-321)

100% correct

Hey all, I'm from Georgia so I hope I can help. I'm also Ruby dev and I heard about this issue from Akira Matsuda's post on Georgian Ruby Community Facebook page. So as I understood you are trying to handle situation when someone calls string manipulation methods on Georgian. Georgian as mentioned, is a single case alphabet so, converting mkhedruli letters to mtavruli ones upon calling .upcase is going against the OOP intuition. upcase, capitalize, lowercase and other methods like these shouldnt do anything to Georgian Unicode because these methods simply do not apply. Let me know if I missed the point and if the problem is something else.

Lasha, if someone calls mkhedruli.upcase, all letters shall be converted to Mtavruli letter, why do you think it should not do anything?

I was thinking, maybe because it is misleading, foreigners may think that those are real upcase versions of Georgian letters, but it is totally separate alphabet, and most of even Georgians don't know how to recognize them. Maybe call the method "მხედრული".to_mtavruli ?

update: I read the http://www.unicode.org/versions/Unicode11.0.0/ch07.pdf and now I cleared up my confusion with Mtavruli and Asomtavruli, I initially assumed you meant mkhedruli would get converted to Asomtavruli. Was not familiar with Mtavruli at all. I'd still say there's room for debate whether this should become a convention or not. Mtavruli looks like to be a good fit for commercials or newspaper headlines but I wouldn't say that it necessarily should be part of unicode or ruby for that matter. It looks more applicable to css/custom-fonts than backend technologies. disclaimer: I am not a philologist.

Just FYI. Python 3.7 supports Unicode11, and behaves as follows.

$ ./local/bin/python3
Python 3.7.0 (default, Oct 12 2018, 11:29:22)
I don't know Georgian letters at all. (I copy-and-pasted the word (January?) from https://github.com/nodejs/node/issues/22518.)

#14 - 10/12/2018 06:33 AM - webzorg (Lasha Abulashvili)
mame (Yusuke Endoh) wrote:

    Just FYI. Python 3.7 supports Unicode11, and behaves as follows.

    $ ./local/bin/python3
    Python 3.7.0 (default, Oct 12 2018, 11:29:22)
    [GCC 7.3.0] on linux
    Type "help", "copyright", "credits" or "license" for more information.
    >>> 'იანვარი'.upper()
    'ᲘᲐᲜᲕᲐᲘᲕᲐᲜᲘ'
    >>> 'იანვარი'.title()
    'იანვარი'

I downloaded python 3.7, did the same and my output was ᲘᲐᲜᲕᲐᲘᲕᲐᲜᲘ, these characters didn't show up either in my terminal nor in browser. how do I check what are they supposed to be? Cannot look it up here as well https://unicodelookup.com/#%E1%B2%98/1.

update: I could find the letters here, and I can confirm it is mtavruli (all upper case versions) https://www.unicode.org/charts/PDF/U1C90.pdf.

#15 - 10/12/2018 10:31 AM - duerst (Martin Dürst)
mame (Yusuke Endoh)
mame (Yusuke Endoh) wrote:

    Just FYI. Python 3.7 supports Unicode11, and behaves as follows.

    $ ./local/bin/python3
    Python 3.7.0 (default, Oct 12 2018, 11:29:22)
    [GCC 7.3.0] on linux
    Type "help", "copyright", "credits" or "license" for more information.
    >>> 'იანვარი'.upper()
    'ᲘᲐᲜᲕᲐᲘᲕᲐᲜᲘ'
    >>> 'იანვარი'.title()
    'იანვარი'

Many thanks for checking Python. The results make sense given the Unicode data, and align with my current proposal. Can you please try 'იანვარი'.upper().title()? I'm really interested in what result Python produces is that case. A straightforward implementation would produce Იანვარი, but as I said above, I'm not sure this is acceptable.

#16 - 10/12/2018 10:40 AM - duerst (Martin Dürst)
Hello Lasha, George,

Many thanks for your comments. Your input is very much appreciated!

webzorg (Lasha Abulashvili) wrote:

    I downloaded python 3.7, did the same and my output was ᲘᲐᲜᲕᲐᲘᲕᲐᲜᲘ, these characters didn't show up either in my terminal nor in browser.

Yes, characters new in Unicode 11.0 will not be supported yet in many fonts. If you hear about a font that supports Unicode 11 MTAVRULI (I'm writing this in all upper-case so I always remember what it is), please tell us.

On an Unicode mailing list, there was some suggestion to have a temporary option that allows not to produce MTAVRULI until people have upgraded their fonts. But it's difficult to know when people will have upgraded (different people will be earlier or later), and many other characters may also not display in all environments.

    how do I check what are they supposed to be? Cannot look it up here as well https://unicodelookup.com/#%E1%B2%98/1.
Yes, the best way to check is to look at it with a browser (or other tool) that shows the character numbers. I just checked, and Firefox shows the characters as small boxes with hex numbers inside. Then one can use the Unicode charts at the above link to cross-check. Unfortunately, other browsers I have checked (IE and Chrome) only show empty boxes or boxes with question marks.

#17 - 10/12/2018 10:50 AM - webzorg (Lasha Abulashvili)

duerst (Martin Dürst) wrote:

mame (Yusuke Endoh)

mame (Yusuke Endoh) wrote:

Just FYI. Python 3.7 supports Unicode11, and behaves as follows.

$ ./local/bin/python3
Python 3.7.0 (default, Oct 12 2018, 11:29:22)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 'იანვარი'.upper()
'ᲘᲐᲜᲕᲐ阒'  # this is correct (upper)
>>> 'იანვარი'.title()
'იანვარი'  # this is correct (title)

Many thanks for checking Python. The results make sense given the Unicode data, and align with my current proposal.

Can you please try 'იანვარი'.upper().title()? I'm really interested in what result Python produces in that case. A straightforward implementation would produce 'Იანვარი', but as I said above, I'm not sure this is acceptable.

Yes that produced 'Იანვარი', which I would agree with you, is going against the language rules (I haven't seen such usage of Mtavruli font anywhere, ever).

#18 - 10/12/2018 11:11 AM - duerst (Martin Dürst)

webzorg (Lasha Abulashvili) wrote:

Giia (George Pheikrishvili) wrote:

Lasha, if someone calls mkhedruli.upcase, all letters shall be converted to Mtavruli letter, why do you think it should not do anything?

I was thinking, maybe because it is misleading, foreigners may think that those are real upcase versions of Georgian letters, but it is totally separate alphabet, and most of even Georgians don't know how to recognize them.

I was surprised when I read that. I don't read any Georgian, but I have looked at the mkhedruli and MTAVRULI charts, and I wouldn't have problems reading one of them if I knew the other.

Maybe call the method "მჭხდრული".to_mtavruli ?

At the Ruby developers’ meeting on Wednesday in Tokyo, somebody mentioned that the situation with mkhedruli and MTAVRULI has some parallels with Hiragana and Katakana in Japanese (one of the similarities is that it would be extremely odd to start a word with one of these, and then continue with the other). But we don't have any String#to_hiragana or String#to_katakana method in Ruby yet.

update: I read the http://www.unicode.org/versions/Unicode11.0.0/ch07.pdf and now I cleared up my confusion with Mtavruli and Asomtavruli, I initially assumed you meant mkhedruli would get converted to Asomtavruli.

Ah, I see. Asomtavruli indeed looks quite a bit different, but I understand that it's mostly historical.

Was not familiar with Mtavruli at all. I'd still say there's room for debate whether this should become a convention or not. Mtavruli looks like to be a good fit for commercials or newspaper headlines

Yes. But so is UPPER CASE for Latin, Cyrillic,...

but I wouldn't say that it necessarily should be part of unicode or ruby for that matter. It looks more applicable to css/custom-fonts than backend technologies. disclaimer: I am not a philologist.
It was apparently handled by custom fonts for a long time. And there was quite a long discussion in Unicode and ISO about how to handle it. The conclusion was that it should be added to Unicode.

Here are pointers to some of the documents in that discussion:
https://www.unicode.org/L2/L2017/17199-n4827-mtavruli.pdf (this is in both Georgian and English)
http://www.unicode.org/wg2/docs/n4827-mtavruli.pdf (same, ISO version)
http://www.unicode.org/wg2/docs/n4776-mtavruli-support.pdf (letter from Minister of Education and Science of Georgia in support)
http://www.unicode.org/wg2/docs/n4707-georgian.pdf (contains some actual examples)

Given that MTAVRULI is now in Unicode, Ruby has to handle it somehow. I'm not sure we can find a solution that makes everybody happy, but we want to make sure we don't do it completely wrong. So any further feedback is appreciated!

#19 - 10/12/2018 11:13 AM - duerst (Martin Dürst)
webzorg (Lasha Abulashvili) wrote:

duerst (Martin Dürst) wrote:

mame (Yusuke Endoh)

Can you please try 'იანვარი'.upper().title()? I'm really interested in what result Python produces is that case. A straightforward implementation would produce ჱანვარი, but as I said above, I'm not sure this is acceptable.

Yes that produced ჱანვარი, which I would agree with you, is going against the language rules (I haven't seen such usage of Mtavruli font anywhere, ever).

Many thanks for checking! such usage apparently has existed (see Fig. 1/2 of http://www.unicode.org/wg2/docs/n4707-georgian.pdf), but that was more than 100 years ago, so we probably better try to avoid it.

#20 - 10/13/2018 12:05 AM - mame (Yusuke Endoh)
Interesting. Python does not always satisfy a property: s.lower().title() == s.upper().title.

>>> s = "იანვარი"
>>> s.lower().title() == s.upper().title()
False

I agree with this if this result is natural for Georgian. But if not, I'd like to keep the intuitive property. Or, is there already any counterexample against s.uppercase.capitalize == s.lowercase.capitalize?

#21 - 10/13/2018 12:21 AM - mame (Yusuke Endoh)
Okay, znz (Kazuhiro NISHIYAMA) told me that the property is already unsatisfied. I don't object.

s = "s\u00DF"; [s.downcase.capitalize, s.upcase.capitalize]
=> ["Sß", "Sss"]

#22 - 10/13/2018 01:27 AM - Alan.X (Alan Benxton)
duerst (Martin Dürst) wrote:

Yes, characters new in Unicode 11.0 will not be supported yet in many fonts. If you hear about a font that supports Unicode 11 MTAVRULI (I'm writing this in all upper-case so I always remember what it is), please tell us.

Hello, you can use these BPG Dejavu fonts: https://bpgfonts.wordpress.com/2018/09/07/gnu-gpl-license-grant-to-linux-distributors/

Mtavruli letters are only used to emphasize whole words; there is no initial-letter capitalization in Georgian. Therefore, the Mkhedruli letters do not have Mtavruli letters as their titlecase, but are explicitly mapped to themselves. This means that in Ruby, mkhedruli.capitalize would be a no-op although mkhedruli.upcase would convert to Mtavruli letters.

Yes, it's correct. Capitalizing the first letter of a word is not expected in modern Georgian. For example in CSS, text-transform:capitalize property has no effect, because ICU's title-casing API leaves alone Georgian lowercase letters.

#23 - 12/10/2018 04:41 AM - duerst (Martin Dürst)
- Status changed from Feedback to Closed

I have implemented this so that String#capitalize on Georgian text produces all-lowercase results.

This means that formally, 'string'.capitalize can be defined as 'string'.downcase.capitalize. This means that for Georgian text, s.downcase.capitalize == s.upcase.capitalize (but as noted above (https://bugs.ruby-lang.org/issues/14839#note-21), not for Ruby in general). Also, this means Ruby
behaves different from Python.

The reason for this is mainly that e.g. "This is an IMPORTANT sentence.".capitalize results in "This is an important sentence.", and this should work in Georgian, too.

A second (and secondary) reason is that implementation was actually easier, because only the first character of the string needs separate behavior for Georgian.