Ruby master - Feature #18554
Move unicode_normalize to a default gem
01/27/2022 04:49 PM - headius (Charles Nutter)

Status: Open
Priority: Normal
Assignee:
Target version:

Description
Could we move the rest of unicode_normalize to a default gem?

The recent updates were mostly updating the Unicode tables, which a user might want to be able to update in an existing Ruby installation. Additionally, this is one of the few stdlib we have to copy into JRuby from the CRuby repository; it would be easier for both if we just pulled in a default gem.

History
#1 - 01/27/2022 05:03 PM - jeremyevans0 (Jeremy Evans)
- Backport deleted (2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN, 3.1: UNKNOWN)
- Tracker changed from Bug to Feature

#2 - 01/28/2022 01:03 AM - shyouhei (Shyouhei Urabe)
Just leaving my :+1: to this idea; not sure how difficult though.

#3 - 01/28/2022 09:33 AM - duerst (Martin Dürst)
Just a few comments, not sure I have thought everything through completely.

One of the motivations for implementing unciode_normalize in pure Ruby was to make it easy for other Ruby implementations to use this code, so from this viewpoint, if it helps JRuby, that would be a plus.

However, contrary to stuff that is in gems now, unicode_normalize part and parcel of the String class, without needing require. It just is placed in lib/ because there was no other, better, place for it. There is already some mechanism for automatic requiring, see function unicode_normalize_common in

Regarding Unicode versions, if somebody wants to change to a specific Unicode version different from what a Ruby version offers, then this would apply not only to unicode_normalize, but also, and probably much more importantly, to regular expressions. But regular expressions are quite tightly linked with Ruby itself, and it would probably be difficult to disentangle them, because it's not much Ruby and a lot of C.

Also, the updating of Unicode versions uses the same logic to get the necessary data for both regular expressions and unicode_normalize, so if unicode_normalize would be separated into a gem, that part might have to be duplicated, creating additional work on this end.

#4 - 01/31/2022 05:51 PM - headius (Charles Nutter)

duerst (Martin Dürst) Thank you for spelling that out. I figured there are some additional nuances to this, since the unicode tables are also in C code and used internally by many parts of Ruby. In that regard, it is at least much easier for us to import the unicode_normalize tables since they are just a matter of copying Ruby code from CRuby to JRuby.

I would like to understand what would break if a user updated unicode_normalize to a newer (or older) version of Unicode than what is natively supported in CRuby. Is this situation likely to break something?

Along a similar line, could the unicode tables in C code also be moved out to a default gem and be made upgradable without rebuilding CRuby? If this were the case, we would contribute code to generate the same tables in Java and have full CRuby/JRuby support for upgraging Unicode tables from a gem.

Granted that these tables are probably used at the lowest levels of CRuby, during boot and otherwise, so I am unsure what other mine fields lie along this path.