Ruby - Bug #2095

Oniguruma No Longer Understands Unihan Characters

09/13/2009 09:21 AM - runpaint (Run Paint Run Run)

Status:	Closed	
Priority:	Normal	
Assignee:	naruse (Yui NARUSE)	
Target version:		
ruby -v:	ruby 1.9.2dev (2009-09-11) [i686-linux]	**Backport:**

Description

=begin
As Oniguruma was undocumented, the recent update was based mainly on guesswork. While working on a Unicode library to create an exhaustive test suite I noticed that the update introduced a serious regression. We based the update on UnicodeData.txt and Scripts.txt, but as the former omits Unihan characters their properties are no longer recognized. To fix this we can have tool/enc-unicode.rb parse Unihan.txt (or, rather, the files to which it is divided over as of Unicode 5.2). However, I'd prefer instead to update the script to use the new XML dump Unicode has made available. This is comprehensive and the simpler, standardized file format means parsing bugs are far less likely. In addition it makes it easier to expand our Unicode support in the feature simply by selecting additional attributes. Unfortunately, both approaches preclude storing the data file(s) in SVN (as we currently do with UnicodeData.txt and Scripts.txt) because the Unihan.txt file alone is 28MB uncompressed. (The XML dump is, of course, even bigger).

In the next 24 hours I will update the script to download the latest XML dump and parse it.
=end

History

#1 - 09/13/2009 09:40 AM - nobu (Nobuyoshi Nakada)

- Category set to M17N

- Assignee set to naruse (Yui NARUSE)

=begin

=end

#2 - 09/14/2009 09:31 AM - runpaint (Run Paint Run Run)

=begin
Having re-written said script I discovered that my initial analysis was wrong; there is no bug. This ticket can be closed. I apologize. :-/
=end

#3 - 09/14/2009 10:43 AM - naruse (Yui NARUSE)

- Status changed from Open to Closed

=begin
ok I close this.

Anyway I thougt UnicodeData.txt and Scripts.txt are also large.
So those source data shouldn't be bundled with Ruby, and download by enc-unicode.rb when it runs and uset them.

So you can use XML dump :-)
=end

#4 - 09/14/2009 03:41 PM - duerst (Martin Dürst)

=begin

On 2009/09/13 9:21, Run Paint Run Run wrote:

> Bug #2095: Oniguruma No Longer Understands Unihan Characters
> http://redmine.ruby-lang.org/issues/show/2095
>
> Author: Run Paint Run Run
> Status: Open, Priority: High

ruby -v: ruby 1.9.2dev (2009-09-11) [i686-linux]

As Oniguruma was undocumented, the recent update was based mainly on guesswork.

We based the update on UnicodeData.txt and Scripts.txt,

UnicodeData.txt since ages contains two-line entries such as

3400;<CJK Ideograph Extension A, First>;Lo;0;L;;;;;N;;;;;
4DB5;<CJK Ideograph Extension A, Last>;Lo;0;L;;;;;N;;;;;

or

4E00;<CJK Ideograph, First>;Lo;0;L;;;;;N;;;;;
9FC3;<CJK Ideograph, Last>;Lo;0;L;;;;;N;;;;;

or

AC00;<Hangul Syllable, First>;Lo;0;L;;;;;N;;;;;
D7A3;<Hangul Syllable, Last>;Lo;0;L;;;;;N;;;;;
D800;<Non Private Use High Surrogate, First>;Cs;0;L;;;;;N;;;;;
DB7F;<Non Private Use High Surrogate, Last>;Cs;0;L;;;;;N;;;;;
DB80;<Private Use High Surrogate, First>;Cs;0;L;;;;;N;;;;;
DBFF;<Private Use High Surrogate, Last>;Cs;0;L;;;;;N;;;;;
DC00;<Low Surrogate, First>;Cs;0;L;;;;;N;;;;;
DFFF;<Low Surrogate, Last>;Cs;0;L;;;;;N;;;;;
E000;<Private Use, First>;Co;0;L;;;;;N;;;;;
F8FF;<Private Use, Last>;Co;0;L;;;;;N;;;;;

These are indications of any of the following:

1. All the characters in the respective range have the same property
   (e.g. 'Lo' for CJK Ideographs)
2. Certain properties essentially don't apply (e.g. Surrogates are 'L',
   but for Ruby, they should not exist, and certainly not match in Regexps)
3. Properties or other relevant data should be generated algorithmically
   (e.g. Character Names for Ideographs and Hangul, normalization
   (de)compositions for Hangul,...)

In my experience, it is best to handle each of these specific ranges explicitly in a script such as yours, and to throw an error (and use a patch to fix it) when a new range is encountered, because a) new such ranges are added rarely (currently, there are only 10), and b) it is impossible to predict which of the above three cases applies.

Regards, Martin.

but as the former omits Unihan characters their properties are no longer recognized. To fix this we can have tool/enc-unicode.rb parse Unihan.txt (or, rather, the files to which it is divided over as of Unicode 5.2). However, I'd prefer instead to update the script to use the new XML dump Unicode has made available. This is comprehensive and the simpler, standardized file format means parsing bugs are far less likely. In addition it makes it easier to expand our Unicode support in the feature simply by selecting additional attributes. Unfortunately, both approaches preclude storing the data file(s) in SVN (as we currently do with UnicodeData.txt and Scripts.txt) because the Unihan.txt file alone is 28MB uncompresse!
d. (The XML dump is, of course, even bigger).

In the next 24 hours I will update the script to download the latest XML dump and parse it.

---

http://redmine.ruby-lang.org

--
#-# Martin J. Dürst, Professor, Aoyama Gakuin University
#-# http://www.sw.it.aoyama.ac.jp mailto:duerst@it.aoyama.ac.jp

=end

#5 - 09/14/2009 03:52 PM - duerst (Martin Dürst)

=begin

On 2009/09/14 10:43, Yui NARUSE wrote:

Issue #2095 has been updated by Yui NARUSE.

> Status changed from Open to Closed
>
> ok I close this.
>
> Anyway I thougt UnicodeData.txt and Scripts.txt are also large.

Please note that this means that implementations will take the newest Unicode version when compiled; this may not work if older Ruby versions (such as 1.9.1) do not want to follow Unicode versions automatically.

This is fine with me as I support following the newest final Unicode versions, but you argued the other way a few weeks ago, and we haven't heard back yet on this issue from Yugui.

Also, it makes it more difficult to check a beta version of Unicode on a 'beta' (or trunk) version of Ruby unless this test is limited to individual (human) compilers.

Regards, Martin.

> So those source data shouldn't be bundled with Ruby, and download by enc-unicode.rb when it runs and uset them.
>
> ## So you can use XML dump :-)
>
> http://redmine.ruby-lang.org/issues/show/2095
>
> ---
>
> http://redmine.ruby-lang.org

--
#-# Martin J. Dürst, Professor, Aoyama Gakuin University
#-# http://www.sw.it.aoyama.ac.jp mailto:duerst@it.aoyama.ac.jp

=end

#6 - 09/14/2009 04:09 PM - runpaint (Run Paint Run Run)

=begin

> Anyway I thougt UnicodeData.txt and Scripts.txt are also large.

They're nothing compared to the full XML dump (~130MB). ;-)

> So you can use XML dump :-)

Well given that I've written the script now, I guess it does no harm to keep it. Maybe we can look at changing over once we have tests.

> In my experience, it is best to handle each of these specific ranges explicitly in a script such as yours, and to throw an error (and use a patch to fix it) when a new range is encountered, because a) new such ranges are added rarely (currently, there are only 10), and b) it is impossible to predict which of the above three cases applies.

Thanks. :-) This was part of the reason I wanted to use the XML dump, because I suspected it would make this kind of thing easier. (I'm learning Unicode as I go ;-)).

> Please note that this means that implementations will take the newest Unicode version when compiled; this may not work if older Ruby versions (such as 1.9.1) do not want to follow Unicode versions automatically.

To clarify, the property table is not regenerated on compilation; we manually update it when we want to synchronize with a new Unicode version. :-)
=end