Ruby trunk - Feature #15317

How to deal with obsolete property values in Unicode 11.0.0

11/18/2018 09:35 AM - duerst (Martin Dürst)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:	2.6	

Description

http://www.unicode.org/versions/Unicode11.0.0/#Migration contains the following:

> Four Grapheme_Cluster_Break and Word_Break classes have become obsolete and are no longer used: E_Base, E_Modifier, Glue_After_Zwj, and E_Base_GAZ. Those values are still part of the enumeration of the property values, because stability constraints prevent removal of enumerated property values, even if obsolete; however, these are no longer assigned to any characters, and are no longer referred to explicitly by any rules in the algorithms.

For Ruby, we have to decide how to support (or not) these property values. The main choices are to throw an error or to just not match anything. The later seems preferable for backwards compatibility, but the relevant file (https://www.unicode.org/Public/UCD/latest/ucd/auxiliary/GraphemeBreakProperty.txt) does not mention these property values anymore.

I'm currently contacting other Unicode experts to find out whether there's some machine readable data for obsolete properties.

Your input is appreciated.

Related issues:

Blocks Ruby trunk - Feature #14802: Update Unicode data to Unicode Version 11...	**Closed**

History

#1 - 11/18/2018 09:36 AM - duerst (Martin Dürst)

- Blocks Feature #14802: Update Unicode data to Unicode Version 11.0.0 added

#2 - 11/18/2018 10:43 AM - shevegen (Robert A. Heiler)

Could a warning be issued as well, at the least for a transition period?

On a side note, does anyone happen to know how perl5/perl6 and python handle these situations? Perhaps if what they do makes sense, we could have a consistent behaviour in this regard across the languages (but only if it makes sense what they do in this context).

#3 - 11/19/2018 10:57 AM - duerst (Martin Dürst)

shevegen (Robert A. Heiler) wrote:

> Could a warning be issued as well, at the least for a transition period?

I warning might make sense, but then we would get into the question of whether we need a warning for those cases w
here property values changed (because obsoleting a property value essentially is the same as changing the value of the property for those characters that previously had the now obsoleted property value).

Often, property values only change for new characters (a defined character usually has different properties form an unassigned code point), which would not need a warning, and edge cases. That could lead to many superfluous warnings. It would also be quite difficult to implement, because the implementation would have to look at two or more sets of property files in parallel.

#4 - 11/20/2018 10:22 AM - duerst (Martin Dürst)

Some pointers obtained from an Unicode-internal discussion:

- All (including past) property values are available from the Relax NG schema for UCD in XML at http://www.unicode.org/reports/tr42/tr42-23.rnc, linked off https://www.unicode.org/reports/tr42/.
- PropertyAliases.txt lists all the properties, and PropertyValueAliases.txt provides lists of property values for enumerated values. We already download these files as part of the Ruby make process.

- Hiragana_or_Katakana is an old obsolete script property, which currently leads to an error with 'abc' =~ /\p{hiragara_or_katakana}/'

#5 - 11/22/2018 07:51 AM - duerst (Martin Dürst)

duerst (Martin Dürst) wrote:

> - Hiragana_or_Katakana is an old obsolete script property, which currently leads to an error with 'abc' =~ /\p{hiragara_or_katakana}/'

A more recent example: 'abc' =~ /\p{Grapheme_Cluster_Break=E_Modifier}/'. This will work with Unicode 10.0.0, but may produce an error with Unicode 11.0.0.

The data for this property is available at https://www.unicode.org/Public/UCD/latest/ucd/auxiliary/GraphemeBreakProperty.txt (latest, i.e. 11.0.0) or the versioned https://www.unicode.org/Public/10.0.0/ucd/auxiliary/GraphemeBreakProperty.txt.

Property values that are not used anymore do not show up in this data file. So E_Modifier is in the 10.0.0 version, but not in the latest version. This results in Grapheme_Cluster_Break=E_Modifier not showing up in enc/unicode/11.0.0/name2ctype.h, which produces an error.

#6 - 11/22/2018 07:54 AM - duerst (Martin Dürst)

The opinions at the committer meeting were tending towards producing an error or a warning, because this would make it possible to find places that need to be rewritten to produce whatever may have been the desired result.

The discussion on the Unicode expert mailing list on the other hand tended towards not producing an error.

#7 - 11/22/2018 08:21 AM - naruse (Yui NARUSE)

duerst (Martin Dürst) wrote:

> duerst (Martin Dürst) wrote:
>
> > - Hiragana_or_Katakana is an old obsolete script property, which currently leads to an error with 'abc' =~ /\p{hiragara_or_katakana}/'
>
> A more recent example: 'abc' =~ /\p{Grapheme_Cluster_Break=E_Modifier}/'. This will work with Unicode 10.0.0, but may produce an error with Unicode 11.0.0.
>
> The data for this property is available at https://www.unicode.org/Public/UCD/latest/ucd/auxiliary/GraphemeBreakProperty.txt (latest, i.e. 11.0.0) or the versioned https://www.unicode.org/Public/10.0.0/ucd/auxiliary/GraphemeBreakProperty.txt.
>
> Property values that are not used anymore do not show up in this data file. So E_Modifier is in the 10.0.0 version, but not in the latest version. This results in Grapheme_Cluster_Break=E_Modifier not showing up in enc/unicode/11.0.0/name2ctype.h, which produces an error.

/\p{Grapheme_Cluster_Break=E_Modifier}/ is specially introduced for /\X/.
But the source of \X, Unicode Text Segmentation (https://unicode.org/reports/tr29/) but whose definition is changed.
Therefore the compatibility about this is not important so much.

So just error seems ok.

#8 - 12/07/2018 10:06 AM - duerst (Martin Dürst)

- Status changed from Open to Closed

naruse (Yui NARUSE) wrote:

> /\p{Grapheme_Cluster_Break=E_Modifier}/ is specially introduced for /\X/.
> But the source of \X, Unicode Text Segmentation (https://unicode.org/reports/tr29/) but whose definition is changed.
> Therefore the compatibility about this is not important so much.
>
> So just error seems ok.

Ok. /\p{Grapheme_Cluster_Break=E_Modifier}/ now produces an error. If we get any bug reports, we can still revisit this issue (actually, better open a new one).