NAME

Unicode::Collate::Locale - Linguistic tailoring for DUCET via Unicode::Collate

SYNOPSIS

```
use Unicode::Collate::Locale;

#construct
$Collator = Unicode::Collate::Locale->
new(locale => $locale_name, %tailoring);

#sort
@sorted = $Collator->sort(@not_sorted);

#compare
$result = $Collator->cmp($a, $b); # returns 1, 0, or -1.
```

Note: Strings in `@not_sorted`, `$a` and `$b` are interpreted according to Perl's Unicode support. See perlunicode(1), perluniintro(1), perlunitut(1), perlunifaq(1), utf8. Otherwise you can use `preprocess` (cf. `Unicode::Collate` or should decode them before.

DESCRIPTION

This module provides linguistic tailoring for it taking advantage of `Unicode::Collate`

Constructor

The `new` method returns a collator object.

A parameter list for the constructor is a hash, which can include a special key `locale` and its value (case-insensitive) standing for a Unicode base language code (two or three-letter). For example, `Unicode::Collate::Locale->new(locale => 'FR')`returns a collator tailored for French.

`$locale_name` may be suffixed with a Unicode script code (four-letter), a Unicode region code, a Unicode language variant code. These codes are case-insensitive, and separated with `'_'` or `'-'`. E.g. `en_US` for English in USA, `az_Cyrl` for Azerbaijani in the Cyrillic script, `es_ES_traditional` for Spanish in Spain (Traditional).

If `$locale_name` is not available, fallback is selected in the following order:

```
1. language with a variant code
2. language with a script code
3. language with a region code
4. language
5. default
```

Tailoring tags provided by `Unicode::Collate` are allowed as long as they are not used for `locale` support. Esp. the `table` tag is always untailorable, since it is reserved for DUCET.

However `entry` is allowed, even if it is used for `locale` support, to add or override mappings.

E.g. a collator for French, which ignores diacritics and case difference (i.e. level 1), with reversed case ordering and no normalization.

```
Unicode::Collate::Locale->new(
level => 1,
locale => 'fr',
upper_before_lower => 1,
normalization => undef
)
```

Overriding a behavior already tailored by `locale` is disallowed if such a tailoring is passed to `new()`.

```
Unicode::Collate::Locale->new(
locale => 'da',
upper_before_lower => 0, # causes error as reserved by 'da'
)
```

However `change()` inherited from `Unicode::Collate` allows such a tailoring that is reserved by `locale`. Examples:

```
new(locale => 'ca')->change(backwards => undef)
new(locale => 'da')->change(upper_before_lower => 0)
new(locale => 'ja')->change(overrideCJK => undef)
```

Methods

`Unicode::Collate::Locale` is a subclass of `Unicode::Collate` and methods other than `new` are inherited from `Unicode::Collate`

Here is a list of additional methods:

`$Collator->getlocale`

Returns a language code accepted and used actually on collation. If linguistic tailoring is not provided for a language code you passed (intensionally for some languages, or due to the incomplete implementation), this method returns a string `'default'` meaning no special tailoring.

`$Collator->locale_version`

(Since Unicode::Collate::Locale 0.87) Returns the version number (perhaps `/\d\.\d\d/`) of the locale, as that of *Locale/*.pl*.

Note: *Locale/*.pl* that a collator uses should be identified by a combination of return values from `getlocale` and `locale_version`.

A list of tailorable locales

```
locale name description
------------------------------------------------------------------
af Afrikaans
ar Arabic
as Assamese
az Azerbaijani (Azeri)
be Belarusian
bg Bulgarian
bn Bengali
bs Bosnian
bs_Cyrl Bosnian in Cyrillic (tailored as Serbian)
ca Catalan
cs Czech
cy Welsh
da Danish
de__phonebook German (umlaut as 'ae', 'oe', 'ue')
ee Ewe
eo Esperanto
es Spanish
es__traditional Spanish ('ch' and 'll' as a grapheme)
et Estonian
fa Persian
fi Finnish (v and w are primary equal)
fi__phonebook Finnish (v and w as separate characters)
fil Filipino
fo Faroese
fr French
gu Gujarati
```

```
ha Hausa
haw Hawaiian
hi Hindi
hr Croatian
hu Hungarian
hy Armenian
ig Igbo
is Icelandic
ja Japanese [1]
kk Kazakh
kl Kalaallisut
kn Kannada
ko Korean [2]
kok Konkani
ln Lingala
lt Lithuanian
lv Latvian
mk Macedonian
ml Malayalam
mr Marathi
mt Maltese
nb Norwegian Bokmal
nn Norwegian Nynorsk
nso Northern Sotho
om Oromo
or Oriya
pa Punjabi
pl Polish
ro Romanian
ru Russian
sa Sanskrit
se Northern Sami
si Sinhala
si__dictionary Sinhala (U+0DA5 = U+0DA2,0DCA,0DA4)
sk Slovak
sl Slovenian
sq Albanian
sr Serbian
sr_Latn Serbian in Latin (tailored as Croatian)
sv Swedish (v and w are primary equal)
sv__reformed Swedish (v and w as separate characters)
ta Tamil
te Telugu
th Thai
tn Tswana
to Tonga
tr Turkish
uk Ukrainian
ur Urdu
vi Vietnamese
wae Walser
wo Wolof
yo Yoruba
zh Chinese
```

```
zh__big5han Chinese (ideographs: big5 order)
zh__gb2312han Chinese (ideographs: GB-2312 order)
zh__pinyin Chinese (ideographs: pinyin order) [3]
zh__stroke Chinese (ideographs: stroke order) [3]
zh__zhuyin Chinese (ideographs: zhuyin order) [3]
----------------------------------------------------------------
```

Locales according to the default UCA rules include chr (Cherokee), de (German), en (English), ga (Irish), id (Indonesian), it (Italian), ka (Georgian), ms (Malay), nl (Dutch), pt (Portuguese), st (Southern Sotho), sw (Swahili), xh (Xhosa), zu (Zulu).

Note

[1] ja: Ideographs are sorted in JIS X 0208 order. Fullwidth and halfwidth forms are identical to their regular form. The difference between hiragana and katakana is at the 4th level, the comparison also requires `(variable => 'Non-ignorable')`, and then `katakana_before_hiragana` has no effect.

[2] ko: Plenty of ideographs are sorted by their reading. Such an ideograph is primary (level 1) equal to, and secondary (level 2) greater than, the corresponding hangul syllable.

[3] zh__pinyin, zh__stroke and zh__zhuyin: implemented alt='short', where a smaller number of ideographs are tailored.

Note: 'pinyin' is in latin, 'zhuyin' is in bopomofo.

INSTALL

Installation of `Unicode::Collate::Locale` requires *Collate/Locale.pm*, *Collate/Locale/*.pm*, *Collate/CJK/*.pm* and *Collate/allkeys.txt*. On building, `Unicode::Collate::Locale` doesn't require any of *data/*.txt*, *gendata/**, and *mklocale*. Tests for `Unicode::Collate::Locale` are named *t/loc_*.t*.

CAVEAT

tailoring is not maximum

Even if a certain letter is tailored, its equivalent would not always tailored as well as it. For example, even though W is tailored, fullwidth W (`U+FF37`), W with acute (`U+1E82`), etc. are not tailored. The result may depend on whether source strings are normalized or not, and whether decomposed or composed. Thus `(normalization => undef)` is less preferred.

AUTHOR

The Unicode::Collate::Locale module for perl was written by SADAHIRO Tomoyuki, <SADAHIRO@cpan.org>. This module is Copyright(C) 2004-2013, SADAHIRO Tomoyuki. Japan. All rights reserved.

This module is free software; you can redistribute it and/or modify it under the same terms as Perl itself.

SEE ALSO

Unicode Collation Algorithm - UTS #10
<http://www.unicode.org/reports/tr10/>

The Default Unicode Collation Element Table (DUCET)
<http://www.unicode.org/Public/UCA/latest/allkeys.txt>

Unicode Locale Data Markup Language (LDML) - UTS #35
<http://www.unicode.org/reports/tr35/>

CLDR - Unicode Common Locale Data Repository
<http://cldr.unicode.org/>

Unicode::Collate
Unicode::Normalize