

# SWI-Prolog Unicode library

Jan Wielemaker  
VU University, Amsterdam  
The Netherlands  
E-mail: `J.Wielemaker@cs.vu.nl`

January 9, 2017

## Abstract

This package wraps [\[utf8proc\]](http://www.public-software-group.org/utf8proc)<http://www.public-software-group.org/utf8proc> unicode routines. This library provides the four unicode normalization forms (NFC, NFD, NFKC, NFKD) as well as access to the Unicode character properties.

## Contents

<b>1</b>	<b>library(unicode): Unicode string handling</b>	<b>3</b>
<b>2</b>	<b>License</b>	<b>6</b>

## 1 library(unicode): Unicode string handling

See also <http://www.public-software-group.org/utf8proc>

This library is a wrapper around the [utf8proc](#) library, providing information about Unicode code-points and performing operations (mappings) on Unicode atoms. The central predicate is `unicode_map/3`, mapping a Unicode atom to another Unicode atom using a sequence of operations. The predicates `unicode_nfd/2`, `unicode_nfc/2`, `unicode_nfkd/2` and `unicode_nfkc/2` implement the four standard Unicode normalization forms.

Lump handling:

```
U+0020      <-- all space characters (general category Zs)
U+0027      '   <-- left/right single quotation mark U+2018..2019,
                  modifier letter apostrophe U+02BC,
                  modifier letter vertical line U+02C8
U+002D      -   <-- all dash characters (general category Pd),
                  minus U+2212
U+002F      /   <-- fraction slash U+2044,
                  division slash U+2215
U+003A      :   <-- ratio U+2236
U+003C      <   <-- single left-pointing angle quotation mark U+2039,
                  left-pointing angle bracket U+2329,
                  left angle bracket U+3008
U+003E      >   <-- single right-pointing angle quotation mark U+203A,
                  right-pointing angle bracket U+232A,
                  right angle bracket U+3009
U+005C      \   <-- set minus U+2216
U+005E      ^   <-- modifier letter up arrowhead U+02C4,
                  modifier letter circumflex accent U+02C6,
                  caret U+2038,
                  up arrowhead U+2303
U+005F      _   <-- all connector characters (general category Pc),
                  modifier letter low macron U+02CD
U+0060      `   <-- modifier letter grave accent U+02CB
U+007C      |   <-- divides U+2223
U+007E      ~   <-- tilde operator U+223C
```

**unicode\_map(+In, -Out, +Options)**

[det]

Perform unicode normalization operations. *Options* is a list of operations. Defined operations are:

**stable**

Unicode Versioning Stability has to be respected.

**compat**

Compatibility decomposition (i.e. formatting information is lost)

**compose**

Return a result with composed characters.

**decompose**

Return a result with decomposed characters.

**ignore**

Strip "default ignorable characters"

**rejectna**

Return an error, if the input contains unassigned code points.

**nlf2ls**

Indicating that NLF-sequences (LF, CRLF, CR, NEL) are representing a line break, and should be converted to the unicode character for line separation (LS).

**nlf2ps**

Indicating that NLF-sequences are representing a paragraph break, and should be converted to the unicode character for paragraph separation (PS).

**nlf2lf**

Indicating that the meaning of NLF-sequences is unknown.

**stripcc**

Strips and/or converts control characters. NLF-sequences are transformed into space, except if one of the NLF2LS/PS/LF options is given. HorizontalTab (HT) and FormFeed (FF) are treated as a NLF-sequence in this case. All other control characters are simply removed.

**casefold**

Performs unicode case folding, to be able to do a case-insensitive string comparison.

**charbound**

Inserts 0xFF bytes at the beginning of each sequence which is representing a single grapheme cluster (see UAX#29).

**lump**

(e.g. HYPHEN U+2010 and MINUS U+2212 to ASCII "-"). (See module header for details.) If NLF2LF is set, this includes a transformation of paragraph and line separators to ASCII line-feed (LF).

**stripmark**

Strips all character markings (non-spacing, spacing and enclosing) (i.e. accents) NOTE: this option works only with `compose` or `decompose`.

**unicode\_nfd(+In, -Out)**

[det]

Characters are decomposed by canonical equivalence.

**unicode\_nfc(+In, -Out)**

[det]

Characters are decomposed and then recomposed by canonical equivalence. It is possible for the result to be a different sequence of characters than the original.

See also [http://en.wikipedia.org/wiki/Unicode\\_equivalence#Normal\\_forms](http://en.wikipedia.org/wiki/Unicode_equivalence#Normal_forms)

**unicode\_nfkd(+In, -Out)**

[det]

Characters are decomposed by compatibility equivalence.

**unicode\_nfkc**(+In, -Out) [det]  
Characters are decomposed by compatibility equivalence, then recomposed by canonical equivalence.

**unicode\_property**(?Char, ?Property) [nondet]  
True if *Property* is defined for *Char*. *Property* is a term Name(Value). Defined property-names are:

**category**(atom)  
Unicode code category of *Char*. This is one of Cc, Cf, Cn, Co, Cs, Ll, Lm, Lo, Lt, Lu, Mc, Me, Mn, Nd, Nl, No, Pc, Pd, Pe, Pf, Pi, Po, Ps, Sc, Sk, Sm, So, Zl, Zp, Zs. When testing, a single letter stands for all its subcategories. E.g. to test form a letter, you can use

```
unicode_property(C, category('L'))
```

**combining\_class**(integer)

**bidirectional\_class**(atom)

**decomposition\_type**(atom)

**decomposition\_mapping**(list(code))

**bidirectional\_mirrored**(bool)

**uppercase\_mapping**(code)

**lowercase\_mapping**(code)

**titlecase\_mapping**(code)

**combining\_index1**(code)

**combining\_index2**(code)

**composition\_exclusion**(bool)

**ignorable**(bool)

**control\_boundary**(bool)

**extend**(bool)

`casefold_mapping(list(code))`

**To be done** Complete documentation

## 2 License

Copyright (c) 2009 Public Software Group e. V., Berlin, Germany

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This software distribution contains derived data from a modified version of the Unicode data files. The following license applies to that data:

### **COPYRIGHT AND PERMISSION NOTICE**

Copyright (c) 1991-2007 Unicode, Inc. All rights reserved. Distributed under the Terms of Use in <http://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that (a) the above copyright notice(s) and this permission notice appear with all copies of the Data Files or Software, (b) both the above copyright notice(s) and this permission notice appear in associated documentation, and (c) there is clear notice in each modified Data File or in the Software as well as in the documentation associated with the Data File(s) or Software that the data or software has been modified.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

Unicode and the Unicode logo are trademarks of Unicode, Inc., and may be registered in some jurisdictions. All other trademarks and registered trademarks mentioned herein are the property of their respective owners.

## Index

unicode\_map/3, 3  
unicode\_nfc/2, 4  
unicode\_nfd/2, 4  
unicode\_nfkc/2, 5  
unicode\_nfkd/2, 4  
unicode\_property/2, 5