

```
;; bo-ewts.mim -- Tibetan input method with EWTS
;; Copyright (C) 2007, 2008, 2009
;; National Institute of Advanced Industrial Science and
Technology (AIST)
;; Registration Number H15PR0112
;; Copyright (C) 2010 Hugues MOISY <hugues.moisy@gmail.com>

;; This file is part of the m17n database; a sub-part of the m17n
;; library.

;; The m17n library is free software; you can redistribute it and/or
;; modify it under the terms of the GNU Lesser General Public
License
;; as published by the Free Software Foundation; either version 2.1
of
;; the License, or (at your option) any later version.

;; The m17n library is distributed in the hope that it will be
useful,
;; but WITHOUT ANY WARRANTY; without even the implied warranty of
;; MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
;; Lesser General Public License for more details.

;; You should have received a copy of the GNU Lesser General Public
;; License along with the m17n library; if not, write to the Free
;; Software Foundation, Inc., 51 Franklin Street, Fifth Floor,
;; Boston, MA 02110-1301, USA.

(input-method bo ewts)

(description "Tibetan input method based on EWTS.
This implementation is based on THDL Extended Wylie Transliteration
Scheme
Version 2.0
<http://www.thdl.org/collections/langling/ewts/ewts.php>.")
```

```
(title "ཀ")
```

```
(variable
```

```
  (precomposed
```

```
    (_"Flag to tell whether or not to generate precomposed characters.
```

```
If 1, generate precomposed characters if available
```

```
(e.g. \ཀ\"(U+0F76).
```

```
If 0, generate only decomposed characters (e.g. \ཀ\ " (U+0FB2
```

```
U+0F80).")
```

```
  0 0 1))
```

```
(map
```

```
;; This map is used also for subjoined consonants.
```

```
(consonant
```

```
  ("k" "ཀ")
```

```
  ("kh" "ཁ")
```

```
  ("g" "ག")
```

```
  ("gh" "ཁྲ")
```

```
  ("ng" "ང")
```

```
  ("c" "ཅ")
```

```
  ("ch" "ཆ")
```

```
  ("j" "ཇ")
```

```
  ("ny" "ཉ")
```

```
  ("T" "ཊ")
```

```
  ("Th" "ཋ")
```

```
  ("D" "ཌ")
```

```
  ("Dh" "ཌྷ")
```

```
  ("N" "ཎ")
```

```
  ("t" "ཏ")
```

```
  ("th" "ཐ")
```

```
  ("d" "ཌ")
```

```
  ("dh" "ཌྷ")
```

```
  ("n" "ཎ")
```

```
  ("p" "པ")
```

```
  ("ph" "ཕ")
```

```
  ("b" "བ")
```

("bh" "འབྲོག་པ་")  
("m" "མཚོན་པ་")  
("ts" "ཚེ་མོན་པ་")  
("tsh" "མཚོན་པ་")  
("dz" "མཚོན་པ་")  
("dzh" "མཚོན་པ་")  
("w" "མཚོན་པ་")  
("zh" "མཚོན་པ་")  
("z" "མཚོན་པ་")  
(" ' " "མཚོན་པ་")  
("y" "མཚོན་པ་")  
("r" "མཚོན་པ་")  
("l" "མཚོན་པ་")  
("sh" "མཚོན་པ་")  
("Sh" "མཚོན་པ་")  
("s" "མཚོན་པ་")  
("h" "མཚོན་པ་")  
("a" "མཚོན་པ་")

)

;; Standard Tibetan Stacks listed at:

;; <<http://www.thdl.org/collections/langling/tibstacks.html>>

(standard-stack

("kl" "མཚོན་པ་")  
("kr" "མཚོན་པ་")  
("kw" "མཚོན་པ་")  
("ky" "མཚོན་པ་")  
("khr" "མཚོན་པ་")  
("khw" "མཚོན་པ་")  
("khy" "མཚོན་པ་")  
("gl" "མཚོན་པ་")  
("gr" "མཚོན་པ་")  
("grw" "མཚོན་པ་")  
("gw" "མཚོན་པ་")  
("gy" "མཚོན་པ་")

("cw" "ꠘꠘ")  
("nyw" "ꠘꠗ")  
("tr" "ꠘꠗ")  
("tw" "ꠘꠗ")  
("thr" "ꠘꠗꠘ")  
("dr" "ꠘꠗ")  
("drw" "ꠘꠗꠘ")  
("dw" "ꠘꠗ")  
("pr" "ꠘꠗ")  
("py" "ꠘꠗ")  
("phr" "ꠘꠗꠘ")  
("phy" "ꠘꠗ")  
("phyw" "ꠘꠗꠘ")  
("bl" "ꠘꠗ")  
("br" "ꠘꠗ")  
("by" "ꠘꠗ")  
("mr" "ꠘꠗ")  
("my" "ꠘꠗ")  
("tsw" "ꠘꠗꠘ")  
("tshw" "ꠘꠗꠘꠘ")  
("zhw" "ꠘꠗꠘꠘ")  
("zl" "ꠘꠗꠘꠘ")  
("zw" "ꠘꠗꠘꠘ")  
("rb" "ꠘꠗꠘ")  
("rd" "ꠘꠗꠘ")  
("rdz" "ꠘꠗꠘꠘ")  
("rg" "ꠘꠗꠘ")  
("rgw" "ꠘꠗꠘꠘ")  
("rgy" "ꠘꠗꠘꠘ")  
("rj" "ꠘꠗꠘ")  
("rk" "ꠘꠗꠘ")  
("rky" "ꠘꠗꠘꠘ")  
("rl" "ꠘꠗꠘ")  
("rm" "ꠘꠗꠘ")  
("rmy" "ꠘꠗꠘꠘ")  
("rn" "ꠘꠗꠘ")

("rng" "ᠷᠨᠭ")  
("rny" "ᠷᠨᠶ")  
("rt" "ᠷᠲ")  
("rts" "ᠷᠲᠰ")  
("rtsw" "ᠷᠲᠰᠤᠬᠤ")  
("rw" "ᠷᠠᠮ")  
("lb" "ᠯᠪ")  
("lc" "ᠯᠴ")  
("ld" "ᠯᠳ")  
("lg" "ᠯᠭ")  
("lh" "ᠯᠬ")  
("lj" "ᠯᠵ")  
("lk" "ᠯᠬ")  
("lng" "ᠯᠨᠭ")  
("lp" "ᠯᠫ")  
("lt" "ᠯᠲ")  
("shr" "ᠰᠬᠷ")  
("shw" "ᠰᠬᠤᠬ")  
("sb" "ᠰᠪ")  
("sbr" "ᠰᠪᠷ")  
("sby" "ᠰᠪᠶ")  
("sd" "ᠰᠳ")  
("sg" "ᠰᠭ")  
("sgr" "ᠰᠭᠷ")  
("sgy" "ᠰᠭᠶ")  
("sk" "ᠰᠬ")  
("skr" "ᠰᠬᠷ")  
("sky" "ᠰᠬᠶ")  
("sl" "ᠰᠯ")  
("sm" "ᠰᠮ")  
("smr" "ᠰᠮᠷ")  
("smy" "ᠰᠮᠶ")  
("sn" "ᠰᠨ")  
("sng" "ᠰᠨᠭ")  
("snr" "ᠰᠨᠷ")  
("sny" "ᠰᠨᠶ")



(special-subjoined

```
("h" (cond ((= precomposed 0) "ḥ")
            ((= @-1 ?ḥ) (delete @-1 "ḥ"))
            ((= @-1 ?ḥ̣) (delete @-1 "ḥ̣"))
            ((= @-1 ?ḥ̇) (delete @-1 "ḥ̇"))
            ((= @-1 ?ḥ̣̇) (delete @-1 "ḥ̣̇"))
            (1 "ḥ")))
("Sh" (cond ((= precomposed 0) "Ṣ")
            ((= @-1 ?Ṣ) (delete @-1 "Ṣ"))
            (1 "Ṣ")))
("W" "Ṣ")
("Y" "Ṣ")
("R" "Ṣ")
("Z" "Ṣ"))
```

(vowel

```
("a" "Ṷ")
("i" "Ṷ")
("u" "Ṷ")
("e" "Ṷ")
("o" "Ṷ")
("-i" (cond ((= precomposed 0) "Ṷ")
            ((= @-1 ?Ṷ) "Ṷ")
            ((= @-1 ?Ṷ̣) "Ṷ̣")
            (1 "Ṷ")))
("A" "Ṷ")
("I" "Ṷ")
("U" "Ṷ")
("ai" "Ṷ")
("au" "Ṷ")
("uo" "Ṷ")
("ui" "Ṷ")
("ue" "Ṷ")
("r-I" "Ṷ")
("l-I" "Ṷ"))
```

```
("-I" (cond ((= precomposed 0) "᳚")
              ((= @-1 2) "᳛")
              ((= @-1 3) "᳜")
              (1 "᳝"))))
```

(others

;; Numbers

("0" "०")

("1" "१")

("2" "२")

("3" "३")

("4" "ॣ")

("5" "।")

("6" "॥")

("7" "१")

("8" "२")

("9" "ॢ")

;; These half numbers are not in EWTS but for convenience.

("-1" "ॣ")

("-2" "।")

("-3" "॥")

("-4" "१")

("-5" "२")

("-6" "ॢ")

("-7" "ॣ")

("-8" "।")

("-9" "॥")

("-0" "१")

;; Sanskrit-related marks

("H" "᳚")

("M" " ")

("~M" "᳛")

("~M`" "᳜")

("?" " ")

("&" "᳝")

;; Intersyllabic marks



(" " ".")

("\*" " ".")

;; Phrase delimiting marks

("/" " | ")

("//" " ||")

(";" " |.")

("|" " |.")

("!" " |.")

(":" " |.")

("\_" " " ")

("=" " |.")

;; Head marks

("@" " |.")

("#" " |.")

("\$" " |.")

("%" " |.")

;; Paired punctuation marks (brackets)

("<" " |.")

(">" " |.")

("(" " |.")

(")" " |.")

;; Miscellaneous

("~X" " |.")

("X" " |.")

("^" " |.")

(escape

("\\\" (mark BEG) "\\\"))

(non-tibetan

("["))

(unicode

("u" "u" (set MAX-COUNT 4))

("U" "U" (set MAX-COUNT 8))

```
(unescape  
  ("]"))
```

```
(hexadigit  
  ("0" "0")  
  ("1" "1")  
  ("2" "2")  
  ("3" "3")  
  ("4" "4")  
  ("5" "5")  
  ("6" "6")  
  ("7" "7")  
  ("8" "8")  
  ("9" "9")  
  ("A" "A") ("a" "A")  
  ("B" "B") ("b" "B")  
  ("C" "C") ("c" "C")  
  ("D" "D") ("d" "D")  
  ("E" "E") ("e" "E")  
  ("F" "F") ("f" "F"))
```

```
(backspace  
  ((Backspace) (undo)))
```

```
(state  
  (init  
    (consonant (shift after-consonant))  
    (standard-stack (shift after-consonant))  
    (sanskrit (shift after-consonant))  
    (vowel (move @<) "ᳵ" (move @>) (shift after-vowel))  
    (others)  
    (escape (shift dispatch-escape))  
    (non-tibetan (shift non-tibetan))  
    (backspace))
```

```
(after-consonant
```

```
(force-stack (shift consonant-stack))
(break-stack (shift init))
(vowel (shift after-vowel)))
```

```
(consonant-stack
 "ཀ+"
 (special-subjoined (shift after-consonant))
 (consonant (set C @-1) (delete @-1) (add C #x50) (insert C)
            (shift after-consonant))
 (force-stack)
 (nil (pop)))
```

```
(after-vowel
 (force-stack (shift vowel-stack)))
```

```
(vowel-stack
 "ཀ+"
 (vowel (shift after-vowel)))
```

```
(dispatch-escape
 (unicode (shift unicode))
 (nil (delete BEG) (shift init) (unhandle)))
```

```
(unicode
 "ཀ u"
 (t (set COUNT 0) (set UNICODE 0))
 (hexadigit (cond ((> @-1 ?9) (set UNICODE (+ (* UNICODE 16) (- @-1
55))))
             (1 (set UNICODE (+ (* UNICODE 16) (- @-1 48))))))
 (set COUNT (+ COUNT 1))
 (cond ((= COUNT MAX-COUNT) (delete BEG) (insert UNICODE)
      (shift init))))
```

```
(non-tibetan
 "ཀ A"
 (unescape (shift init)))
```

```
(nil (unhandle)))
```

```
;; Local Variables:
```

```
;; mode: lisp
```

```
;; End:
```