NAME

HTPWID – Parse a Hebrew token into substrings and find their total width in points.

SYNOPSIS

CALL HTPWID(TOKEN,LTK,PT, SUBS,NST,WTK,RC)

TOKEN(LTK)	is the CHARACTER*1 token to be parsed
LTK	is the INTEGER*4 number of characters in TOKEN
PT	is the REAL*4 point size at which he token will be printed
SUBS(10)	is the CHARACTER*20 list of substrings found
NST	is the INTEGER*4 number of substrings found
WTK	is the REAL*4 predicted print width of TOKEN in points
RC	is an INTEGER*4 return code; see below

WARNING

The width WTK returned by this routine is only an estimate.

DESCRIPTION

On its first invocation only, this routine calls GETHSH to read the library file ${HOME}/Utility/hebrew.hsh and table its transliterations and hash codes.

The routine parses TOKEN from left to right. If a character is < it is assumed to be the beginning of a transliteration and characters are copied through the closing >. Then TRNIDX is used to find the corresponding byte code, HEBWID is used to find the width of the printed word corresponding to the byte code, and that amount is added to the width of the token.

If a character begins the string \hebpnk{, the entire \hebpnk command is copied. Then STRWID is used to find the width of the printed punctuation mark, and that amount is added to the width of the token.

If a character is a \ that does not begin the string \hebpnk{, it is assumed to be the beginning of a Hebrew letter of the form \vowel{consonant} and characters are copied through the closing }. Then HB2HSH is used to hash the Hebrew, HEBWID is used to find the width of the printed letter corresponding to the byte code, and that amount is added to the width of the token.

Otherwise the character is assumed to be English punctuation. STRWID is used to find its width, and that is added to the width of the token.

UNITS and FILES

0	if compiled with DEBUG/.TRUE./
1	transiently, ${HOME}/Utility/hebrew.hsh

SEE ALSO

HEBWID, which returns the width of a Hebrew string
STRWID, which returns the width of an English string

DIAGNOSTICS

On output these are the possible RC values:

−1	too many subtokens in TOKEN
−2	no closing > on a transliteration
−3	transliteration not found in the dictionary
−4	no closing } on a \vowel{letter} string
−5	HB2HSH returned a hash of more than 1 character
−6	STRWID did not find a character in its tables
−7	HEBWID did not find a hash code in its table
0	width successfully estimated
>0	return code from HB2HSH; see its man page

LINKAGE

```
gfortran   source.f   −L${HOME}/lib   −lmisc
```

AUTHOR

Michael Kupferschmid

EXAMPLE

```
      CHARACTER*33 TOKEN/"\hebpnk{,}<hungar’yah>\sheva{bet}"/
      CHARACTER*20 SUBS(10)
      INTEGER*4 RC
      CALL HTPWID(TOKEN,33,12.0, SUBS,NST,WTK,RC)
      WRITE(6,901) RC,WTK
  901 FORMAT(’RC=’,I2,’ WTK=’,F5.1,’ points’)
      DO 1 I=1,NST
           WRITE(6,902) SUBS(I)
  902      FORMAT(A20)
    1 CONTINUE
      STOP
      END
```

This example produced the following output:

```
unix[1] a.out
RC=0
WTK=51.7 points
\hebpnk{,}
<hungar’yah>
\sheva{bet}
unix[2]
```