

**NAME**

HEBJST – Justify left-to-right Hebrew in right-to-left lines.

**SYNOPSIS**

```
cat text.ltr | ${HOME}/bin/exe/hebjst [pts] > text.rtl
```

**DESCRIPTION**

If the real parameter [pts] is omitted the program prints usage instructions and stops. Otherwise it copies input lines to the output file unchanged except for blocks of text bracketed by flags like this

```
% LTR
text
:
text
% LTR
```

To process the Hebrew text within each such block it reads a line and tokenizes it. It uses HTPWID to decompose each token into subtokens, and assembles the subtokens in reverse order into an output `\heblines{}`, accumulating the predicted print width of the line. When adding another token would make the printed line wider than the page (assumed to be 469.8pt) it prints the current line and begins a new one with that token.

A subtoken that is not a `<transliteration>` or a `\hebpnk{}` command or a `\vowel{consonant}` Hebrew type-setting command is assumed to consist of one or more English punctuation marks. Each of its characters is separately enclosed in a `\hebpnk{}` command and that command is added to the output line in place of the original subtoken character.

A blank line within a flag-bounded block of text signals a paragraph break. The current output line is written, followed by a paragraph space and processing resumes with the next nonblank input line.

**UNITS and FILES**

0	usage information
1	transiently (via HTPWID) <code>\${HOME}/Utility/hebrew.hsh</code>
5	input LaTeX set left-to-right
6	output LaTeX set right-to-left in <code>\heblines{}</code> s

**DIAGNOSTICS**

If HTPWID returns a nonzero return code it is reported and the program stops.

**AUTHOR**

Michael Kupferschmid

**EXAMPLE**

```
unix[1] cat text.ltr | hebjst 24.48 > text.rtl
unix[2]
```

The program copies `text.ltr` into `text.rtl` except for bracketed blocks, which it sets in `\heblines{}`s so that when printed at the nominal size of 24pt (which is actually 24.48pt) the text fits on the page.