# gloss.tex

Macros for vertically aligning words in consecutive sentences.
Version: 1.0 (26 November 1990) (doc: 10 November 1990)


To gloss means to insert explanations or translations of words between the lines of a text. For ease of reading, one generally wants the explained word and the explanation to line up, as in

| | | | | |
|---|---|---|---|---|
| *Dit* | *is* | *een* | *Nederlands* | *voorbeeld.* |
| This | is | a | Dutch | example. |

The macros defined in gloss.tex make it convenient to typeset such glossed sentences. To use the macros, you first have to `\input gloss`*. The main macro for typesetting glossed sentences is `\gloss`. It should be followed by the two sentences, each terminated by a ⟨return⟩. The previous example was typeset with

```
\gloss Dit is een Nederlands voorbeeld.
This is a Dutch example.
```

But instead of printing the glossed sentence, `\gloss` puts the whole construction in an hbox, called `\gline`. To actually insert that box in your text, you use `\box\gline` or `\unhbox\gline`. The text is put in a box to make it easier to handle. For instance, it makes it easy to put extra indentation or an example number in front of the sentence. (The baseline of the hbox is the baseline of the second line in the box.) The difference between `\box\gline` and `\unhbox\gline` is that the latter allows line breaks in the glossed sentence. That is obviously only useful if the glossed sentence is quite long; we will see an example later on.

The `\gloss` macro splits each sentence into words, and then aligns the words. In this context, a word is a sequence of characters separated by spaces. The number of spaces between words is irrelevant, so that you can line up the words in your input as well. Using

```
\gloss Dit  is een Nederlands voorbeeld.
       This is a   Dutch      example.
```

gives exactly the same output as the input given before. However, you should not insert extra ⟨return⟩ symbols. The sentences can have a different number of words. Sometimes one of the two sentences has a word without equivalent in the other sentence. In such cases you can insert an empty word with `{}`. You can also combine several words into a single word, by using grouping. The following example illustrates both.

```
\gloss Dit is een voorbeeldje     in het Nederlands.
       This is a {little example} in {}  Dutch.
```

gives

| | | | | | | |
|---|---|---|---|---|---|---|
| *Dit* | *is* | *een* | *voorbeeldje* | *in* | *het* | *Nederlands.* |
| This | is | a | little example | in | | Dutch. |

Without the grouping "example" would have been lined up with "*in*."

By default the first sentence of a glossed pair is set in italic font (`\it`), the second in roman font (`\rm`). You can change this by, for instance, setting `\let\eachwordone=\rm`, which will set each word in the first sentence in roman font. The font of the second sentence is governed by `\eachwordtwo`. This use of the `\eachword` macros is an instance of a more general mechanism that is explained later. You can insert font changes in the middle of a sentence, but they only apply to the next word.

```
\gloss Dit  is een Nederlands voorbeeld.
       This is a   \sl Dutch      example.
```

will set "Dutch" in slanted font, but "example" in the default font. The `\gloss` macro considers `\sl Dutch` to be a single word, and sets this word in an hbox so that the effect of the font change is limited. `\sl` is not considered a word by itself, because spaces following a macro are always skipped.

* We have made no attempt to LaTeXify gloss.tex, but you can use the macros in LaTeX just as in plain TeX.

You may want to gloss more than a single sentence, maybe a whole paragraph. It would however be
verb verb verb noun noun verb verb
quite inconvenient if you had to follow `\gloss` by a whole paragraph on a single line (by terminating each
verb macro noun noun verb
line with a `%`-sign), followed by a line containing all the glosses. Especially if there are a lot of unglossed
noun noun verb noun verb noun verb noun verb?
(empty) words, it would be hard to get the sentences lined up properly. Fortunately, a paragraph like the
noun verb verb verb noun verb noun
current one can be set quite easily. `\gloss` puts the created sentence in `\gline`, and `\gline` is emptied
noun verb verb verb macro verb verb noun box box verb verb
by using `\box` or `\unhbox`. But if you don't empty `\gline`, the next usage of `\gloss` will append the new
verb verb verb box noun macro verb verb
sentence to the older one. Hence, this paragraph was typeset with
noun noun noun verb verb

```
\gloss You may want to gloss more than a single sentence, maybe
       {} verb verb {} verb  {}   {}   {} {}    noun
\gloss a whole paragraph.
       {} {}   noun
```

⋮ (10 more)

```
\gloss one. Hence, this paragraph was typeset with
       noun {}    {}   noun      verb  verb
\unhbox\gline
```

Of course, a few more things, like the fonts, had to be changed. We discuss that next. Also notice that it takes TEX quite some time to typeset such a paragraph.

Each word from the first line following `\gloss` is set in an hbox as follows

`\hbox{\eachwordone\strut`⟨word⟩`␣}`

The macro `\eachwordone` is by default `\let` equal to `\it`, but can be redefined. The previous paragraph was made with

```
\let\eachwordone=\rm
\def\sixstrut{\vrule height5pt depth1.5pt width0pt }
\def\eachwordtwo{\let\strut=\sixstrut\sixrm}
```

(If `\strut` were not redefined, there would be too much space between the first and the second line.) Furthermore, to obtain a larger separation between the pairs of lines, the `\baselineskip` was set to 22.5 pt (`\baselineskip` governs the spacing between the glossed pairs, the spacing within the pair is governed by the `\strut`s.) Since a word does not contain spaces, and each word is followed by a space, the `\eachword` macros can easily be made to take the word (and the `\strut`) as argument. For instance, the following definition will underline every word in the first line.

```
\def\eachwordone#1␣{\rm\underbar{#1}␣}
```

The space following the word does not disappear at a line break, because it is hidden within an hbox. To hide this extra space between the last word and the right margin, the final thing needed to typeset the previous paragraph was

```
\setbox0=\hbox{␣}\advance\rightskip by-\wd0
```

This is not perfect, because the size of the space depends on the font: there is a verb sticking out slightly.

The word boxes are paired together in a single vbox, and `\gline` consists of a sequence of these vboxes. The vboxes are separated by some glue, equal to `\glossglue`. Without this glue no line breaks would be possible, nor would the spaces between the words be stretchable. `gloss.tex` says `\glossglue=0pt plus2pt minus1pt`. The stretch and shrink components are only relevant if `\unhbox\gline` is used.

Sometimes you would rather not have extra space following a word, for instance when parts of a word are glossed separately: In

*Dit is een voorbeeld- je in het Nederlands.*
This is an example ("little") in Dutch.

it is preferable to have no space following "*voorbeeld-*" if it is not necessary. This can easily be achieved with a macro like

```
\def\+#1␣{#1}
```

which is inserted just before the word that should not have a space: `\+voorbeeld- je`. It can also be done by making the hyphen gobble up the space, as in

```
\let\hyphen=-
\catcode`-=\active
\def-#1␣{\hyphen#1}
```

However, this does not work if there is a word with a hyphen in the middle. To handle that situation, a more sophisticated definition of `-` can check what the next character is, using `\futurelet`. We will not do that here, as it has little to do with glossed sentences.