

# Gloss $\TeX$ 0.3

VOLKAN YAVUZ\*

1997/06/10

## Abstract

Gloss $\TeX$  is a tool for the preparation of glossaries, lists of acronyms or sorted lists in general. It greatly simplifies this task. One or more glossary-definition files serve as databases which contain descriptions of terms. These terms are identified through labels. Based upon labels set into the  $\TeX$ -source, Gloss $\TeX$  determines which entries have to appear in the typeset list. Gloss $\TeX$  uses MAKEINDEX for the sorting of the lists. References to the place where a term appears in the text can be set in the list. A term consists of a label which is used to identify it, an optional item describing the typeset output, an optional longform and the actual text representing it. There are many ways to access each of these fields within the document. It is also possible to generate cross-references to another term.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Purpose . . . . .	2
1.2	History . . . . .	2
1.3	Legalese . . . . .	2
<b>2</b>	<b>Usage</b>	<b>3</b>
2.1	The glossary definition file . . . . .	4
2.2	Invocation . . . . .	5
2.3	Page References . . . . .	6
2.4	Cross-References . . . . .	6
2.5	Gloss $\TeX$ and nomencl . . . . .	7
<b>3</b>	<b>Customizing</b>	<b>7</b>
<b>4</b>	<b>Some Details</b>	<b>8</b>
<b>5</b>	<b>Portability</b>	<b>9</b>

---

\*e-mail: yavuzv@rumms.uni-mannheim.de

<b>6 Acknowledgments</b>	<b>9</b>
<b>7 The implementation</b>	<b>9</b>
7.1 The package <code>glosstex.sty</code> . . . . .	9
7.2 The MAKEINDEX style-file <code>glosstex.mst</code> . . . . .	18

## List of Tables

1 Overview of the different sets of commands. . . . .	4
2 Options controlling the appearance of page references. . . . .	6

## List of Acronyms

# 1 Introduction

## 1.1 Purpose

Gloss $\TeX$  is a tool for the automatic preparation of glossaries, lists of acronyms, nomenclature and sorted lists in general. Based upon the labels set into the  $\TeX$ -source, Gloss $\TeX$  determines which entries from a glossary-definition file have to be processed to generate the list. Gloss $\TeX$  then creates an intermediate file that has to be processed by MAKEINDEX for sorting. The output of MAKEINDEX is then included into the  $\TeX$ -source for typesetting. This whole process can be automated by using a makefile. With each term associated is an item representing the typeset output, an optional long-form if it's an acronym and an optional descriptive text. These elements can all be accessed within the document in many ways.

## 1.2 History

I created Gloss $\TeX$  because there were no tools for the preparation of glossaries that fit my needs. Gloss $\TeX$  is mainly a combination of the features of the packages `acronym`, `nomencl` and `GloTeX`. The commands starting with `\ac` are taken more or less from `acronym` and the way Gloss $\TeX$  handles page references is almost identical to the way `nomencl` does. The use of glossary databases is inspired by `GloTeX`.

## 1.3 Legalese

Gloss $\TeX$  is provided “as is” and comes with absolutely no warranty. It is covered by the GNU General Public License (see the file `COPYING` that comes with this package).

© Volkan Yavuz, 1997

## 2 Usage

The L<sup>A</sup>T<sub>E</sub>X-macros needed by GlossT<sub>E</sub>X have to be included into the source using

```
\usepackage[options]{glosstex}
```

where *options* may be one of `refpage` or `norefpage`. The latter one is the default. These options are described later in section 2.3.

`\glosstex` Whenever you want a term to appear in the glossary, you insert

```
\glosstex[list][pageref-mode]{label}
```

into the text. *label* references the entry and the optional arguments *pageref-mode* and *list* determine the mode for page references and the list the referenced term should appear in. *pageref-mode* works in conjunction with the optional arguments `refpage` and `norefpage` to the package and defaults to `p`. *list* defaults to `glo`.

But you can create *n* lists and make each *label* appear in any *list*, each appearance independent of the other ones. To facilitate the creation of multiple *list*s and especially make it easy to achieve the most likely usage, there are actually two sets of commands. One set starting with `\gl` defaults to *list* (`glo`) (glossary) and *pageref-mode* `p`, the other set starting with `\ac` defaults to *list* (`acr`) (list of acronyms) and *pageref-mode* `n`. The following is the equivalent to `\glosstex`:

`\acronym`

```
\acronym[list][pageref-mode]{label}
```

These two commands can also be called as `\glosstex{*}` and `\acronym{*}` to include all term found in the `.gdf`-files into the corresponding *list*. This is similar to the command `\nocite{*}` in BIB<sub>T</sub><sub>E</sub>X. But do *not* use something like `\gls{*}` (see below).

There is another set of commands that produce both an entry in the list (optional) and typeset output.

`\gls`

`\gls*`

```
\gls[*][list][pageref-mode]{label}
```

`\gls` typesets *item* and produces a list entry (*list* defaults to `glo`). The starred version `\gls*` just produces typeset output without a list-entry. (To be honest, a list-entry is produced, but it never appears in the output.) The starred version implicitly produces entries with *list-mode* `n` (never) while the unstarred versions set *list-mode* to `a` (always). There is no other way of specifying the *list-mode* of an entry.

`\ac`

`\acs`

`\acl`

`\acf`

`\ac*`

`\acs*`

`\acl*`

`\acf*`

```
\ac[*][list][pageref-mode]{label}
```

```
\acs[*][list][pageref-mode]{label}
```

```
\acl[*][list][pageref-mode]{label}
```

```
\acf[*][list][pageref-mode]{label}
```

These all produce typeset output. `\acs` typesets  $\langle item \rangle$ , `\acl` typesets  $\langle long-form \rangle$  and `\acf` typesets  $\langle item \rangle (\langle long-form \rangle)$ . `\ac` works like `\acf` at it's first invocation and like `\acs` on all subsequent ones. Using `\ac` you can make sure that an acronym is always spelled out at least once in your document while safely using the short form whenever possible. The starred versions `\ac*`, `\acs*`, `\acl*` and `\acf*` just produce typeset output. Table 1 gives an overview of all this.

defaults	only list	both	only text	output
<code>(glo)[p]</code>	<code>\glosstex</code>	<code>\gls</code>	<code>\gls*</code>	$\langle item \rangle$
<code>(acr)[n]</code>	<code>\acronym</code>	<code>\ac</code>	<code>\ac*</code>	$\langle item \rangle$ or $\langle item \rangle (\langle long-form \rangle)$
<code>(acr)[n]</code>		<code>\acs</code>	<code>\acs*</code>	$\langle item \rangle$
<code>(acr)[n]</code>		<code>\acl</code>	<code>\acl*</code>	$\langle long-form \rangle$
<code>(acr)[n]</code>		<code>\acf</code>	<code>\acf*</code>	$\langle item \rangle (\langle long-form \rangle)$

Table 1: Overview of the different sets of commands.

## 2.1 The glossary definition file

A glossary-definition file (suffix `.gdf`) is needed which serves as a database for GlossTeX, holding the actual descriptions of all terms. You can have  $m$  `.gdf`-files that contain the definitions to the  $\langle label \rangle$ s you reference in your documents. Entries have the form

```
@entry{ $\langle label \rangle$ [],  $\langle item \rangle$ [],  $\langle long-form \rangle$ []} [ $\langle text \rangle$ ]
```

where  $\langle label \rangle$  is used to identify the entry and  $\langle text \rangle$  may contain any amount of TeX-source, being the actual definition of the item. You should know that  $\langle label \rangle$  is used to construct TeX-macros, so it shouldn't contain funny characters or you will most likely get funny errors. The optional argument  $\langle item \rangle$  describes the appearance of the item in the produced list. If omitted, it defaults to  $\langle label \rangle$ . It can be used when some special form of typesetting is wanted.  $\langle item \rangle$  can contain any TeX-construct, as long as each “{” has a corresponding “}”. Otherwise, GlossTeX will get confused. The same applies to  $\langle long-form \rangle$ , except that it defaults into an empty string if not specified.

This is the `.gdf`-file used for this documentation. Note the use of “~” as the quote-character. Also note that all lines until the first line starting with `@entry{` are ignored. Additionally, all lines starting with “%” are ignored, too. Thus they can serve as comments.

```
1  $\langle *gdf \rangle$ 
2 % -- latex --
3
4 This is a database file for GlossTeX.
5
6 @entry{mst-file, \texttt{.mst}-file} Style file for \MakeIndex{,
7 describing the input and output format of read and written files.
```

8  
9 @entry{gdf-file, \texttt{.gdf}-file} This file is the database file  
10 containing definitions for GlossTeX{ }.  
11  
12 @entry{gxs-file, \texttt{.gxs}-file} Intermediate file produced by  
13 \GlossTeX{ } to be processed by \MakeIndex{ }.  
14  
15 @entry{glx-file, \texttt{.glx}-file} This file contains the sorted  
16 lists, ready to be read by \LaTeX{ }.  
17  
18 @entry{gxg-file, \texttt{.gxg}-file} This is the log-file produced by  
19 the \GlossTeX{ }-run. See also \glxref{glg-file}.  
20  
21 @entry{glg-file, \texttt{.glg}-file} This is the log-file produced by  
22 the \MakeIndex{ }-run. See also \glxref{gxg-file}.  
23  
24 @entry{ASCII, ASCII, American Standard Code for Information Interchange}  
25 A character encoding. See also \glxref{EBCDIC}.  
26  
27 @entry{EBCDIC, EBCDIC, Extended Binary Coded Decimals Interchange Code}  
28 A character encoding exclusively used on mainframes. See also  
29 \glxref{ASCII}.  
30 </gdf>

## 2.2 Invocation

After the first run of L<sup>A</sup>T<sub>E</sub>X, the .aux-file contains all necessary information for the preparation of the glossary. GlossT<sub>E</sub>X is then invoked to read one or more .gdf-files and output all definitions that are referenced in the .aux-file. The output of GlossT<sub>E</sub>X is then processed by MAKEINDEX for sorting.

GlossT<sub>E</sub>X is invoked in a UNIX-like environment using the following command

```
glosstex <aux-file> <gdf-file> [<gdf-file>[...]] [-v[0...5]]
```

This produces 2 files as output, one **gxs-file!** to be input into MAKEINDEX and a log-file with extension .gxg which contains more detailed information. The -v option selects how verbose GlossT<sub>E</sub>X should be when writing the log-file. -v is equal to -v4 and -v2 is the default. -v0 makes GlossT<sub>E</sub>X shut his mouth and only report errors and -v5 makes GlossT<sub>E</sub>X really talkative.

MAKEINDEX has to be invoked in this way

```
makeindex <gxs-file> -o <glx-file> -s <ist-file> [-t <glg-file>]
```

The commands

```
glosstex thesis thesis.gdf master.gdf
makeindex thesis.gxs -o thesis.glx -s glosstex.mst
```

\printglosstex produce the final **glx-file!** which is then included by

`\printglosstex[⟨list⟩][⟨pageref-mode⟩]`

during the next L<sup>A</sup>T<sub>E</sub>X-run. The argument *⟨pageref-mode⟩* supercedes the one given to the entries individually for each *⟨list⟩* and defaults to `[p]`. You can turn on page references unconditionally for each *⟨list⟩* individually by using *⟨pageref-mode⟩* `[a]` and turn it off by using *⟨pageref-mode⟩* `[n]` as argument to `\printglosstex`. See also table 2 for an overview of these options.

Note that, no matter of how many *⟨list⟩*s you produce, there is always exactly *one* `.glx`-file which contains the entries of *all* produced lists. Depending on whether you use *⟨item⟩* or *⟨long-form⟩* in your text or you have cross-references, it may be necessary to run L<sup>A</sup>T<sub>E</sub>X and GlossT<sub>E</sub>X up to 4 times until all references are resolved. Watch out for warnings from `glosstex` during a L<sup>A</sup>T<sub>E</sub>X run. Messages about unresolved *⟨label⟩*s from GlossT<sub>E</sub>X are caused by missing definitions in the `.gdf`-files.

### 2.3 Page References

You may want a reference in the list to the place where the term first appears in the text. This can be done using the optional argument *⟨pageref-mode⟩*. That argument in combination with the option to `\usepackage` controls these references. Table 2 gives an overview of all possible combinations of these 3 arguments. A “×” indicates that a reference is produced.

list entry	refpage			norefpage		
	a	p	n	a	p	n
a	×	×	–	×	×	–
p	×	×	–	×	–	–
n	×	–	–	×	–	–

Table 2: Options controlling the appearance of page references.

One possible usage of this feature: while debugging a document, turn on page references by using the option `refpage` to the package. Every entry included with the modes `[a]` (always) or the default `[p]` (package) will contain a reference. After debugging, remove the option `refpage` and only those entries that were included with mode `[a]` will still have a reference.

### 2.4 Cross-References

`\glxref` It may be useful to use cross-references in entries. Assume you have referenced  
`\glxref*` `\glosstex{ascii}` which describes the term **ASCII!**. You may also want to include **EBCDIC!** as an example for another character encoding. To achieve this, write this into the definition of **ASCII!**

See also `\glxref{ebcdic}`.

and Gloss<sub>TEX</sub> then produces “See also EBCDIC” and also includes the definition for **EBCDIC!** into the same list **ASCII!** appears in. Note that `\glxref` is only available within the *⟨text⟩* argument in the `.gdf`-file since it only makes sense within a *⟨list⟩*. There is also a starred version `\glxref*` that doesn’t produce typeset output.

## 2.5 Gloss<sub>TEX</sub> and nomencl

It is possible to use `nomencl` and Gloss<sub>TEX</sub> in one document without problems.

The following commands show how to deal with documents using both Gloss<sub>TEX</sub> and `nomencl`.

```
latex thesis
glosstex thesis thesis.gdf
makeindex thesis.gxs -o thesis.glx -s glosstex.mst
makeindex thesis.glo -o thesis.gls -s nomencl.ist
latex thesis
```

## 3 Customizing

Gloss<sub>TEX</sub> can be customized by using the file `glosstex.cfg` which is automatically loaded if it is present. The file `glosstex.std` is the default configuration file that is mandatory to Gloss<sub>TEX</sub>. It shows all aspects that are meant to be customized, so let’s discuss it now.

Each term that gets typeset either goes through `\GLX@output@short` or `\GLX@output@long`, depending whether it’s the *⟨item⟩* or *⟨long-form⟩*. These macros each take 3 arguments and get called this way:

```
\GLX@output@short{⟨label⟩}{⟨list⟩}{⟨item⟩}
\GLX@output@long{⟨label⟩}{⟨list⟩}{⟨long-form⟩}
```

`\GLX@output@short` In `\GLX@output@short` we test whether we are in the list of acronyms (`\GLX@acdef@list`) and typeset *⟨item⟩* with caps and small caps if we are. Otherwise, just typeset *⟨item⟩*. An index-entry is also produced, using *⟨label⟩* as the sort-key.

```
31 ⟨*std⟩
32 \newcommand{\GLX@output@short}[3]{%
33 \ifthenelse{\equal{#2}{\GLX@acdef@list}}{\textsc{#3}}{#3}%
34 \index{#1@#3}}
35 \newcommand{\GLX@output@long}[3]{#3}
```

`\glosstexpage` This is used to typeset the page at the end of a definition. It uses `\pagename` so that should be defined elsewhere.

```
36 \newcommand{\glosstexpage}[1]{\nobreak\hfill\emph{\pagename~#1}\nobreak}
```

`\glxgldefault` These macros set the defaults for *⟨list⟩* and *⟨pageref-mode⟩* which the commands starting with `\gl` (except for `\glxref`, of course) and `\ac` are using.

```
\glxacdefault
37 \glxgldefault{glo}{p}
38 \glxacdefault{acr}{n}
```

Each  $\langle list \rangle$  gets embedded into  $\backslash GLX@benv@ \langle list \rangle$  and  $\backslash GLX@eenv@ \langle list \rangle$ , so these macros should provide a reasonable environment. Each line itself is typeset using  $\backslash GLX@item@ \langle list \rangle$  which gets called with 7 arguments.

```

\GLX@benv@glo
\GLX@eenv@glo
\GLX@item@glo
\GLX@benv@acr
\GLX@eenv@acr
\GLX@item@acr
    \GLX@item@ \langle list \rangle { \langle label \rangle } { \langle item \rangle } { \langle long-form \rangle } { \langle text \rangle } { \langle list \rangle } { \langle list-mode \rangle } { \langle page-stuff \rangle }
39 \newcommand{\GLX@benv@glo}{\begin{description}}
40 \newcommand{\GLX@eenv@glo}{\end{description}}
41 \newcommand{\GLX@item@glo}[7]{%
42   \item[#2]\ifx#3\empty\else\emph{#3}\space\fi#4\space#7}
43
44 \newcommand{\GLX@item@acr@label}[1]{\mbox{#1}\dotfill}
45 \newcommand{\GLX@benv@acr}{%
46   \begin{list}{}{}%
47     \renewcommand{\makelabel}{\GLX@item@acr@label}%
48     \setlength{\labelwidth}{7em}%
49     \leftmargin\labelwidth \advance\leftmargin by \labelsep}}
50 \newcommand{\GLX@eenv@acr}{\end{list}}
51 \newcommand{\GLX@item@acr}[7]{%
52   \item[\textsc{#2}]%
53   \ifx#3\empty\else#3\quad\fi\ifx#4\empty\else#4\fi\space#7}

```

$\backslash glossaryname$  These are defined to contain some default strings if they're not already defined  
 $\backslash listacronymname$  ( $\backslash pagename$  e.g. is defined through  $\backslash babel$ ).

```

\pagename
54 \ifx\glossaryname\@undefined
55   \def\glossaryname{Glossary}
56 \fi
57 \ifx\listacronymname\@undefined
58   \def\listacronymname{List of Acronyms}
59 \fi
60 \ifx\pagename\@undefined
61   \def\pagename{page}
62 \fi

```

$\backslash glxheading$  Each  $\langle list \rangle$  starts with an appropriate heading which is defined by

$\backslash glxheading [ \langle list \rangle ] { \langle definition \rangle } .$

```

63 \ifx\chapter\@undefined
64   \glxheading{\section*{\glossaryname}}
65   \glxheading[acr]{\section*{\listacronymname}}
66 \else
67   \glxheading{\chapter*{\glossaryname}}
68   \glxheading[acr]{\chapter*{\listacronymname}}
69 \fi
70 \</std>

```

## 4 Some Details

While reading the  $\backslash .aux$ -file, GlossT<sub>E</sub>X only considers the first appearance of one  $\langle item \rangle$  for each  $\langle list \rangle$ . All subsequent entries are silently ignored. (Almost silently,



because the `.gxx`-file will contain detailed information about this, and more.) But if the first entry says not to produce a page reference and a following one says to do so, then the latter will supercede the former. The same applies if a term is referenced without the option not to generate a list-entry (all commands containing a `*`, e.g. `\gl*`), but a following tells to do so.

While reading one or more `.gdf`-files, only the first definition is used, all other entries are ignored. This fact can be utilised in some way. Assume you have a `master.gdf` which contains general terms and a file `thesis.gdf` which only contains terms that are intended for use in your thesis. Whenever an entry is present in both `.gdf`-files, the one from `thesis.gdf` should be taken. To achieve this, specify `thesis.gdf before master.gdf`.

The `.gxx`-file contains additional information, e.g. when no entry was found for a label.

After GlossTeX is finished, it will print some statistics about read labels, unresolved entries and the like. More detailed information can then be found in the `.gxx`-file.

## 5 Portability

GlossTeX should compile in any ANSI C environment. I have built GlossTeX on Linux 2.0.0 with gcc 2.7.2, GNU make 3.7.4 and libc 5.2.17. It also compiled out-of-the box on DOS using DJGPP 2.6.3. There is also an OS/2 port on CTAN in `support/glosstex/binaries/os2`. If you have successfully ported GlossTeX on some other platform (like VMS or Macintosh) please contact me, so I can enhance GlossTeX.

Additionally, see the file `TODD` in this package for known bugs (also called features) and not yet implemented features (also called bugs).

## 6 Acknowledgments

I would like to thank these people who have contributed to the development of GlossTeX:

STEFAN A. DEUTSCHER, MICHAEL FRIENDLY

## 7 The implementation

### 7.1 The package `glosstex.sty`

First, we identify ourselves.

```
71 <*package>
72 \NeedsTeXFormat{LaTeX2e}[1996/12/01]
73 \ProvidesPackage{glosstex}
74 [\filedate\space\fileversion\space GlossTeX package]
```

The ifthen package is needed for some \ifthenelse-constructs.

```

75 \RequirePackage{ifthen}
76 \newif\if@GLX@pagerefmode

\GlossTeX This is used to typeset a nice logo for this package, which looks like “GlossTEX”.
77 \newcommand{\GlossTeX}{Gloss\TeX}

\GlossTeXPage Call \glosstexpage depending on #2 (pageref-mode), \if@GLX@pagerefmode
and \GLX@theglosstex@pagerefmode.
78 \newcommand{\GlossTeXPage}[2]{\ignorespaces%
79 \ifthenelse{\boolean{@GLX@pagerefmode}}%
80   {% repage true
81     \ifthenelse{%
82       \equal{\GLX@theglosstex@pagerefmode}{a}}{\glosstexpage{#2}}{%
83       \ifthenelse{\equal{\GLX@theglosstex@pagerefmode}{n}}{}{%
84         \ifthenelse{\equal{#1}{a}}{\glosstexpage{#2}}{}%
85         \ifthenelse{\equal{#1}{p}}{\glosstexpage{#2}}{}%
86         \ifthenelse{\equal{#1}{n}}{}{}}}%
87   {% repage false
88     \ifthenelse{%
89       \equal{\GLX@theglosstex@pagerefmode}{a}}{\glosstexpage{#2}}{%
90       \ifthenelse{\equal{\GLX@theglosstex@pagerefmode}{n}}{}{%
91         \ifthenelse{\equal{#1}{a}}{\glosstexpage{#2}}{}%
92         \ifthenelse{\equal{#1}{p}}{}{}%
93         \ifthenelse{\equal{#1}{n}}{}{}}}%
94   }%

Now declare the options this package accepts and execute the standard option
norepage.
95 \DeclareOption{repage}{\@GLX@pagerefmodetrue}
96 \DeclareOption{norepage}{\@GLX@pagerefmodefalse}
97 \ExecuteOptions{norepage}
98 \ProcessOptions

\GlossTeXNull This one is used as a dummy to work around a feature in MAKEINDEX.
99 \newcommand{\GlossTeXNull}[1]{}

\glosstex \glosstex[list][[pageref-mode]]{label} just produces a list-entry.
100 \DeclareRobustCommand\glosstex{\GLX@glosstex}
101 \newcommand{\GLX@glosstex}{%
102   \@ifnextchar[{\GLX@@glosstexs}{%
103     \@ifnextchar({\GLX@@glosstexr}
104       {\GLX@@@glosstexs[\GLX@gldef@page] (\GLX@gldef@list)}}}
105 \newcommand{\GLX@@glosstexs}{%
106 \def\GLX@@glosstexs[#1]{%
107   \@ifnextchar({\GLX@@@glosstexs[#1]}
108     {\GLX@@@glosstexs[#1] (\GLX@gldef@list)}}}
109 \newcommand{\GLX@@glosstexr}{%
110 \def\GLX@@glosstexr[#1]{%

```

```

111 \@ifnextchar[{\GLX@@glosstexr(#1)}
112 {\GLX@@glosstexs[\GLX@gldef@page](#1)}
113 \newcommand{\GLX@@glosstexs}{}
114 \def\GLX@@glosstexs[#1](#2)#3{\GLX@do@glosstex{#3}{#1}{#2}{a}}
115 \newcommand{\GLX@@glosstexr}{}
116 \def\GLX@@glosstexr(#1)[#2]#3{\GLX@do@glosstex{#3}{#2}{#1}{a}}

\acronym \acronym[(<list>)] [<pageref-mode>]{<label>} just produces a list-entry.
117 \DeclareRobustCommand\acronym{\GLX@acronym}
118 \newcommand{\GLX@acronym}{%
119 \@ifnextchar[{\GLX@@acronyms}{%
120 \@ifnextchar({\GLX@@acronymr}{\GLX@@acronyms[n](acr)}}}
121 \newcommand{\GLX@@acronyms}{}
122 \def\GLX@@acronyms[#1]{%
123 \@ifnextchar({\GLX@@acronyms[#1]}{\GLX@@acronyms[#1](acr)}}
124 \newcommand{\GLX@@acronymr}{}
125 \def\GLX@@acronymr(#1){%
126 \@ifnextchar[{\GLX@@acronymr(#1)}{\GLX@@acronyms[n](#1)}
127 \newcommand{\GLX@@acronyms}{}
128 \def\GLX@@acronyms[#1](#2)#3{\GLX@do@glosstex{#3}{#1}{#2}{a}}
129 \newcommand{\GLX@@acronymr}{}
130 \def\GLX@@acronymr(#1)[#2]#3{\GLX@do@glosstex{#3}{#2}{#1}{a}}

\gls
\gls* 131 \DeclareRobustCommand\gls{%
132 \@ifstar{\GLX@glsS}{\GLX@gls}
133 \newcommand{\GLX@gls}{%
134 \@ifnextchar[{\GLX@@glss}{%
135 \@ifnextchar({\GLX@@glsr}
136 {\GLX@@glss[\GLX@gldef@page](\GLX@gldef@list)}}}
137 \newcommand{\GLX@@glss}{}
138 \def\GLX@@glss[#1]{%
139 \@ifnextchar({\GLX@@@glss[#1]}{\GLX@@@glss[#1](\GLX@gldef@list)}}
140 \newcommand{\GLX@@glsr}{}
141 \def\GLX@@glsr(#1){%
142 \@ifnextchar[{\GLX@@@glsr(#1)}{\GLX@@@glss[\GLX@gldef@page](#1)}
143 \newcommand{\GLX@@@glss}{}
144 \def\GLX@@@glss[#1](#2)#3{%
145 \GLX@do@glosstex{#3}{#1}{#2}{a}\GLX@do@acs{#3}{#2}}
146 \newcommand{\GLX@@@glsr}{}
147 \def\GLX@@@glsr(#1)[#2]#3{%
148 \GLX@do@glosstex{#3}{#2}{#1}{a}\GLX@do@acs{#3}{#1}}
149 \newcommand{\GLX@glsS}{%
150 \@ifnextchar[{\GLX@glsSs}{%
151 \@ifnextchar({\GLX@glsSr}
152 {\GLX@glsSs[\GLX@gldef@page](\GLX@gldef@list)}}}
153 \newcommand{\GLX@glsSs}{}
154 \def\GLX@glsSs[#1]{%
155 \@ifnextchar({\GLX@glsSs[#1]}{\GLX@glsSs[#1](\GLX@gldef@list)}}
156 \newcommand{\GLX@glsSr}{}

```

```

157 \def\GLX@@glsSr(#1){%
158   \@ifnextchar[{\GLX@@glsSr(#1)}{\GLX@@glsSs[\GLX@gldef@page](#1)}}
159 \newcommand{\GLX@@glsSs}{%
160 \def\GLX@@glsSs[#1](#2)#3{%
161   \GLX@do@glosstex{#3}{#1}{#2}{n}\GLX@do@acs{#3}{#2}}
162 \newcommand{\GLX@@glsSr}{%
163 \def\GLX@@glsSr(#1)[#2]#3{%
164   \GLX@do@glosstex{#3}{#2}{#1}{n}\GLX@do@acs{#3}{#1}}

\ac
\acs* 165 \DeclareRobustCommand\ac{%
166   \@ifstar{\GLX@acS}{\GLX@ac}}
167 \newcommand{\GLX@ac}{%
168   \@ifnextchar[{\GLX@@acs}{%
169     \@ifnextchar[{\GLX@@acr}
170       {\GLX@@@acs[\GLX@acdef@page](\GLX@acdef@list)}}}
171 \newcommand{\GLX@@acs}{%
172 \def\GLX@@acs[#1]{%
173   \@ifnextchar[{\GLX@@@acs[#1]}{\GLX@@@acs[#1](\GLX@acdef@list)}}}
174 \newcommand{\GLX@@acr}{%
175 \def\GLX@@acr(#1){%
176   \@ifnextchar[{\GLX@@@acr(#1)}{\GLX@@@acs[\GLX@acdef@page](#1)}}}
177 \newcommand{\GLX@@@acs}{%
178 \def\GLX@@@acs[#1](#2)#3{%
179   \GLX@do@glosstex{#3}{#1}{#2}{a}\GLX@do@ac{#3}{#2}}
180 \newcommand{\GLX@@@acr}{%
181 \def\GLX@@@acr(#1)[#2]#3{%
182   \GLX@do@glosstex{#3}{#2}{#1}{a}\GLX@do@ac{#3}{#1}}}
183 \newcommand{\GLX@acS}{%
184   \@ifnextchar[{\GLX@@acSs}{%
185     \@ifnextchar[{\GLX@@acSr}
186       {\GLX@@@acSs[\GLX@acdef@page](\GLX@acdef@list)}}}
187 \newcommand{\GLX@@acSs}{%
188 \def\GLX@@acSs[#1]{%
189   \@ifnextchar[{\GLX@@@acSs[#1]}{\GLX@@@acSs[#1](\GLX@acdef@list)}}}
190 \newcommand{\GLX@@acSr}{%
191 \def\GLX@@acSr(#1){%
192   \@ifnextchar[{\GLX@@@acSr(#1)}{\GLX@@@acSs[\GLX@acdef@page](#1)}}}
193 \newcommand{\GLX@@@acSs}{%
194 \def\GLX@@@acSs[#1](#2)#3{%
195   \GLX@do@glosstex{#3}{#1}{#2}{n}\GLX@do@ac{#3}{#2}}
196 \newcommand{\GLX@@@acSr}{%
197 \def\GLX@@@acSr(#1)[#2]#3{%
198   \GLX@do@glosstex{#3}{#2}{#1}{n}\GLX@do@ac{#3}{#1}}

\acs
\acs* 199 \DeclareRobustCommand\acs{%
200   \@ifstar{\GLX@acsS}{\GLX@acs}}
201 \newcommand{\GLX@acs}{%
202   \@ifnextchar[{\GLX@@acss}{%

```

```

203 \@ifnextchar({\GLX@@acsr}
204 {\GLX@@@acss[\GLX@acdef@page] (\GLX@acdef@list)}}}
205 \newcommand{\GLX@@acss}{-}
206 \def\GLX@@acss[#1]{%
207 \@ifnextchar({\GLX@@@acss[#1]}{\GLX@@@acss[#1] (\GLX@acdef@list)}}}
208 \newcommand{\GLX@@acsr}{-}
209 \def\GLX@@acsr(#1){%
210 \@ifnextchar[{\GLX@@@acsr(#1)}{\GLX@@@acss[\GLX@acdef@page] (#1)}}}
211 \newcommand{\GLX@@@acss}{-}
212 \def\GLX@@@acss[#1] (#2) #3{%
213 \GLX@do@glosstex{#3}{#1}{#2}{a}\GLX@do@acs{#3}{#2}}
214 \newcommand{\GLX@@@acsr}{-}
215 \def\GLX@@@acsr(#1) [#2] #3{%
216 \GLX@do@glosstex{#3}{#2}{#1}{a}\GLX@do@acs{#3}{#1}}
217 \newcommand{\GLX@acsS}{-}
218 \@ifnextchar[{\GLX@@acsSs}{-}
219 \@ifnextchar({\GLX@@acsSr}
220 {\GLX@@@acsSs[\GLX@acdef@page] (\GLX@acdef@list)}}}
221 \newcommand{\GLX@@acsSs}{-}
222 \def\GLX@@acsSs[#1]{%
223 \@ifnextchar({\GLX@@@acsSs[#1]}{\GLX@@@acsSs[#1] (\GLX@acdef@list)}}}
224 \newcommand{\GLX@@acsSr}{-}
225 \def\GLX@@acsSr(#1){%
226 \@ifnextchar[{\GLX@@@acsSr(#1)}{\GLX@@@acsSs[\GLX@acdef@page] (#1)}}}
227 \newcommand{\GLX@@@acsSs}{-}
228 \def\GLX@@@acsSs[#1] (#2) #3{%
229 \GLX@do@glosstex{#3}{#1}{#2}{n}\GLX@do@acs{#3}{#2}}
230 \newcommand{\GLX@@@acsSr}{-}
231 \def\GLX@@@acsSr(#1) [#2] #3{%
232 \GLX@do@glosstex{#3}{#2}{#1}{n}\GLX@do@acs{#3}{#1}}

```

\acl

```

233 \DeclareRobustCommand\acl{%
234 \@ifstar{\GLX@acls}{\GLX@acl}}
235 \newcommand{\GLX@acl}{-}
236 \@ifnextchar[{\GLX@@acls}{-}
237 \@ifnextchar({\GLX@@aclr}
238 {\GLX@@@acls[\GLX@acdef@page] (\GLX@acdef@list)}}}
239 \newcommand{\GLX@@acls}{-}
240 \def\GLX@@acls[#1]{%
241 \@ifnextchar({\GLX@@@acls[#1]}{\GLX@@@acls[#1] (\GLX@acdef@list)}}}
242 \newcommand{\GLX@@aclr}{-}
243 \def\GLX@@aclr(#1){%
244 \@ifnextchar[{\GLX@@@aclr(#1)}{\GLX@@@acls[\GLX@acdef@page] (#1)}}}
245 \newcommand{\GLX@@@acls}{-}
246 \def\GLX@@@acls[#1] (#2) #3{%
247 \GLX@do@glosstex{#3}{#1}{#2}{a}\GLX@do@acl{#3}{#2}}
248 \newcommand{\GLX@@@aclr}{-}
249 \def\GLX@@@aclr(#1) [#2] #3{%
250 \GLX@do@glosstex{#3}{#2}{#1}{a}\GLX@do@acl{#3}{#1}}

```

```

251 \newcommand{\GLX@ac1S}{%
252   \@ifnextchar[{\GLX@@ac1Ss}{%
253     \@ifnextchar({\GLX@@ac1Sr}
254       {\GLX@@@ac1Ss[\GLX@acdef@page] (\GLX@acdef@list)}}}
255 \newcommand{\GLX@@ac1Ss}{%
256 \def\GLX@@ac1Ss[#1]{%
257   \@ifnextchar({\GLX@@@ac1Ss[#1]}{\GLX@@@ac1Ss[#1] (\GLX@acdef@list)}}}
258 \newcommand{\GLX@@ac1Sr}{%
259 \def\GLX@@ac1Sr(#1){%
260   \@ifnextchar[{\GLX@@@ac1Sr(#1)}{\GLX@@@ac1Ss[\GLX@acdef@page] (#1)}}}
261 \newcommand{\GLX@@@ac1Ss}{%
262 \def\GLX@@@ac1Ss[#1] (#2) #3{%
263   \GLX@do@glosstex{#3}{#1}{#2}{n}\GLX@do@ac1{#3}{#2}}
264 \newcommand{\GLX@@@ac1Sr}{%
265 \def\GLX@@@ac1Sr(#1) [#2] #3{%
266   \GLX@do@glosstex{#3}{#2}{#1}{n}\GLX@do@ac1{#3}{#1}}

```

\acf

```

\acf* 267 \DeclareRobustCommand\acf{%
268   \@ifstar{\GLX@acfS}{\GLX@acf}}
269 \newcommand{\GLX@acf}{%
270   \@ifnextchar[{\GLX@@acfs}{%
271     \@ifnextchar({\GLX@@acfr}
272       {\GLX@@@acfs[\GLX@acdef@page] (\GLX@acdef@list)}}}
273 \newcommand{\GLX@@acfs}{%
274 \def\GLX@@acfs[#1]{%
275   \@ifnextchar({\GLX@@@acfs[#1]}{\GLX@@@acfs[#1] (\GLX@acdef@list)}}}
276 \newcommand{\GLX@@acfr}{%
277 \def\GLX@@acfr(#1){%
278   \@ifnextchar[{\GLX@@@acfr(#1)}{\GLX@@@acfs[\GLX@acdef@page] (#1)}}}
279 \newcommand{\GLX@@@acfs}{%
280 \def\GLX@@@acfs[#1] (#2) #3{%
281   \GLX@do@glosstex{#3}{#1}{#2}{a}\GLX@do@acf{#3}{#2}}
282 \newcommand{\GLX@@@acfr}{%
283 \def\GLX@@@acfr(#1) [#2] #3{%
284   \GLX@do@glosstex{#3}{#2}{#1}{a}\GLX@do@acf{#3}{#1}}
285 \newcommand{\GLX@acfS}{%
286   \@ifnextchar[{\GLX@@acfSs}{%
287     \@ifnextchar({\GLX@@acfSr}
288       {\GLX@@@acfSs[\GLX@acdef@page] (\GLX@acdef@list)}}}
289 \newcommand{\GLX@@acfSs}{%
290 \def\GLX@@acfSs[#1]{%
291   \@ifnextchar({\GLX@@@acfSs[#1]}{\GLX@@@acfSs[#1] (\GLX@acdef@list)}}}
292 \newcommand{\GLX@@acfSr}{%
293 \def\GLX@@acfSr(#1){%
294   \@ifnextchar[{\GLX@@@acfSr(#1)}{\GLX@@@acfSs[\GLX@acdef@page] (#1)}}}
295 \newcommand{\GLX@@@acfSs}{%
296 \def\GLX@@@acfSs[#1] (#2) #3{%
297   \GLX@do@glosstex{#3}{#1}{#2}{n}\GLX@do@acf{#3}{#2}}
298 \newcommand{\GLX@@@acfSr}{%

```

```

299 \def\GLX@@@acfSr(#1)[#2]#3{%
300   \GLX@do@glosstex{#3}{#2}{#1}{n}\GLX@do@acf{#3}{#1}}

\GLX@entry This is the control-sequence written into the .aux-file which is read by GlossTEX.
This macro is defined so that it takes 5 arguments and expands to \relax when
the .aux-file is reread by TEX.
301 \newcommand{\GLX@entry}[5]{\relax}

\GLX@do@glosstex This macro writes lines of the format

      \GLX@entry{<label>}{<pageref-mode>}{<list>}{<list-mode>}{<page>}

into the .aux-file.
302 \newcommand{\GLX@do@glosstex}[4]{%
303   \protected@write\@auxout{%
304     {\string\GLX@entry{#1}{#2}{#3}{#4}{\thepage}}}}

The following commands create macros so that \GLX@term@<label> contains the
short-form of <label> #1, \GLX@term@@<label> contains the <long-form> of <label> #1
and \GLX@term1@<label> expands to x if the <long-form> of <label> #1 has already
been used. This is heavily borrowed from acronymy.

\GLX@do@acs Output the short-form <item> of <label> #1 in <list> #2.
305 \newcommand{\GLX@do@acs}[2]{%
306   \expandafter\ifx\csname GLX@term@#1\endcsname\relax%
307     \textbf{#1!}%
308     \PackageWarning{glosstex}{Term ‘#1’ is not defined}%
309     \expandafter\gdef\csname GLX@term@#1\endcsname{\textbf{#1!}}%
310   \else%
311     \GLX@output@short{#1}{#2}{\csname GLX@term@#1\endcsname}%
312   \fi}

\GLX@do@acl Output the <long-form> of <label> #1.
313 \newcommand{\GLX@do@acl}[2]{%
314   \expandafter\ifx\csname GLX@term@@#1\endcsname\relax%
315     \textbf{#1!}%
316     \PackageWarning{glosstex}{Term ‘#1’ has no long form}%
317     \expandafter\gdef\csname GLX@term@@#1\endcsname{\textbf{#1!}}%
318   \else%
319     \GLX@output@long{#1}{#2}{\csname GLX@term@@#1\endcsname}%
320   \fi}

\GLX@do@acf Output <short> (<long>) of <label> #1 by calling \GLX@do@acs and \GLX@do@acl.
321 \newcommand{\GLX@do@acf}[2]{%
322   \GLX@do@acs{#1}{#2}%
323   \expandafter\ifx\csname GLX@term@@#1\endcsname\relax%
324     \else%
325     \nolinebreak[3]\space(\GLX@do@acl{#1}{#2})%
326   \fi}%

```

`\GLX@do@ac` Either call `\GLX@do@acf` or `\GLX@do@acs`, depending on `\GLX@term1@#1 == x`.

```
327 \newcommand{\GLX@do@ac}[2]{%
328   \expandafter\ifx\csname GLX@term1@#1\endcsname\relax%
329     \GLX@do@acf{#1}{#2}%
330   \expandafter\gdef\csname GLX@term1@#1\endcsname{x}%
331   \else%
332     \GLX@do@acs{#1}{#2}%
333   \fi}
```

`\GLX@newterm` This macro defines `\GLX@term@#1` et. al so that they expand to their appropriate meaning.

```
334 \newcommand{\GLX@newterm}[3]{%
335   \expandafter\gdef\csname GLX@term@#1\endcsname{#2}
336   \if#3\empty%
337   \else\expandafter\gdef\csname GLX@term@#1\endcsname{#3}\fi}
```

`theglosstex` This environment is used for typesetting the lists generated by `GlossTeX`. These are the meanings of the parameters:

```
#1   This <list>
#2   This <pageref-mode>
##1  <label>
##2  <item>
##3  <long-form>
##4  <text>
##5  <list>
##6  <list-mode>
##7  <page-stuff>
```

The whole list is encapsulated into the environment defined by the macros `\GLX@benv@<list>` and `\GLX@eenv@<list>`. These contain `\begin{description}` and `\end{description}` for `<list> == (glo)` as default. Each `\GlossTeXEntry` is only processed if it's `<list>` is identical to the environment's `<list>` and it's `<list-mode> == [a]`. `\GLX@item@<list>` is called for each of these entries.

```
338 \newenvironment{theglosstex}[2]{%
339   \gdef\GLX@theglosstex@list{#1}%
340   \gdef\GLX@theglosstex@pagerefmode{#2}%
341   \csname GLX@benv@#1\endcsname%
```

`\glxref` This one produces an entry in the current `<list>` (stored in `\GLX@theglosstex@list`). ■

```
\glxrefS 342 \DeclareRobustCommand{\glxref}{%
343   \@ifstar{\GLX@glxrefS}{\GLX@glxref}}
344 \newcommand{\GLX@glxref}[1]{%
345   \GLX@do@glosstex{##1}{n}{\GLX@theglosstex@list}{a}%
346   \GLX@do@acs{##1}{\GLX@theglosstex@list}}
347 \newcommand{\GLX@glxrefS}[1]{%
348   \GLX@do@glosstex{##1}{n}{\GLX@theglosstex@list}{a}}

349 \newcommand{\GlossTeXEntry}[7]{%
350   \GLX@newterm{##1}{##2}{##3}%
```



```

351 \write\@auxout{\string\GLX@newterm{##1}{##2}{##3}}%
352 \ifthenelse{\equal{#1}{##5}}{%
353 \ifthenelse{\equal{a}{##6}}{%
354 \csname GLX@item@#1\endcsname%
355 {##1}{##2}{##3}{##4}{##5}{##6}{##7}}{%
356 }{}}{%
357 \csname GLX@eenv@\GLX@theglosstex@list\endcsname}

```

`\printglosstex` The optional arguments are the *<list>* to be typeset and the *<pageref-mode>*. First, `\GLX@heading@<list>` is called (defaults to something like `\section{<glossaryname>}`) to produce a proper heading. Then, if present, the `.glx`-file is included into a `theglosstex` environment.

```

358 \DeclareRobustCommand\printglosstex{\GLX@printglosstex}
359 \newcommand{\GLX@printglosstex}{%
360 \ifnextchar[{\GLX@@printglosstexs}{%
361 \ifnextchar({\GLX@@printglosstexr}
362 {\GLX@@@printglosstexs[p](glo)}}}
363 \newcommand{\GLX@@printglosstexs}{%
364 \def\GLX@@printglosstexs[#1]{%
365 \ifnextchar({\GLX@@@printglosstexs[#1]}
366 {\GLX@@@printglosstexs[#1](glo)}}}
367 \newcommand{\GLX@@printglosstexr}{%
368 \def\GLX@@printglosstexr(#1){%
369 \ifnextchar[{\GLX@@@printglosstexr(#1)}
370 {\GLX@@@printglosstexs[p](#1)}}}
371 \newcommand{\GLX@@@printglosstexs}{%
372 \def\GLX@@@printglosstexs[#1](#2){\GLX@do@printglosstex{#2}{#1}}
373 \newcommand{\GLX@@@printglosstexr}{%
374 \def\GLX@@@printglosstexr(#1)[#2]{\GLX@do@printglosstex{#1}{#2}}
375 \newcommand{\GLX@do@printglosstex}[2]{%
376 \expandafter\ifx\csname GLX@heading@#1\endcsname\relax%
377 \textbf{#1!}%
378 \PackageWarning{glosstex}{Heading for list ‘#1’ is not defined}%
379 \expandafter\gdef\csname GLX@heading@#1\endcsname{\textbf{#1!}}%
380 \else%
381 \csname GLX@heading@#1\endcsname%
382 \IfFileExists{\jobname.glx}{%
383 \begin{theglosstex}{#1}{#2}%
384 \input{\jobname.glx}%
385 \end{theglosstex}}%
386 {\typeout{No file \jobname.glx}}%
387 \fi}

```

`\glxgldefault` The commands set the default *<list>* and *<pageref-mode>* for the commands starting with `\gl` and `\ac`, respectively.

```

388 \newcommand{\glxgldefault}[2]{%
389 \gdef\GLX@gldef@list{#1}%
390 \gdef\GLX@gldef@page{#2}}
391

```

```

392 \newcommand{\glxacdefault}[2]{%
393   \gdef\GLX@acdef@list{#1}%
394   \gdef\GLX@acdef@page{#2}}

```

`\glxheading` This macro sets the headings to be used typesetting a *list*.

```

395 \newcommand{\glxheading}[2][glo]{%
396   \expandafter\gdef\csname GLX@heading@#1\endcsname{#2}}

```

Now include the mandatory standard configuration file `glosstex.std` and an optional `.cfg`-file.

```

397 \InputIfFileExists{glosstex.std}{%
398   \typeout{Using the standard configuration file glosstex.std}}{%
399   \PackageWarning{glosstex}{%
400     File glosstex.std not found. Expect many errors}}
401 \InputIfFileExists{glosstex.cfg}{%
402   \typeout{Using the configuration file glosstex.cfg}}{}
403 \</package>

```

## 7.2 The MakeIndex style-file `glosstex.mst`

It is very important to use an appropriate **mst-file!** for MAKEINDEX to be able to read and write files in the correct format. GlossTeX comes with `glosstex.mst` which has to be used as style-file for MAKEINDEX. The file `glosstex.mst` should not be modified since it is tightly coupled to the features of MAKEINDEX. Note the use of “~” as `quote` to enable german word-ordering using the `-g` option to MAKEINDEX. So if you want a “~” in your `.gdf`-file, you have to type “~”

```

404 \<*mst>
405 keyword      "\\GlossTeXEntry"
406 quote        '~'
407 delim_0      ""
408 group_skip   "\n"
409 indent_length 0
410 indent_space ""
411 item_0       "\n\\GlossTeXEntry"
412 postamble    "\n"
413 preamble     "\n"
414 \</mst>

```

## Glossary