leipzig package documentation*

Natalie Weber
natalie.a.weber@gmail.com

2013/05/26

Abstract

The leipzig package provides a set of macros for standard glossing abbreviations, with options to create new ones. They are mnemonic (e.g. `\Acc{}` for accusative, abbreviated ACC). They can be used alone or on top of the glossaries package for easy indexing and glossary printing.

Contents

*This document corresponds to leipzig v1.0, dated 2013/05/26.

1 Introduction[1]

The leipzig package pre-defines (in leipzig.tex) many of the most common gloss abbreviations (essentially equivalent to the appendix to the Leipzig Glossing Rules [1]). They are pre-defined to save the end-user time, and also to encourage standardization. They are called with macros like `\Nom{}`, `\Acc{}`, and `\Dat{}`, which typeset NOM, ACC, and DAT, respectively. These macros are short, mnemonic, and stand out visually when editing interlinear gloss texts (IGTs) in the source code. Additional macros are provided to define new abbreviations and to set global display parameters which affect all glosses. When used in conjunction with the glossaries package, the abbreviation macros are indexed in a glossary. These can then be printed in a list, table, or inline glossary; an example of the latter is in a footnote on this page.

[1]The leipzig and glossaries package were used to automatically index and print the following gloss abbreviations used in this documentation: 1 = first person , 2 = second person , 3 = third person, ACC = accusative, COP = copula, DAT = dative , DU = dual, INC = inclusive , INS = instrumental, NOM = nominative, PL = plural, SG = singular, VB = verbalizer . **Note**: the spaces before punctuation items in this list are abnormal and due to a clash with the doc class used to typset package documentation.

2 Installation and declaration

2.1 Basic package

Download the leipzig package from CTAN and save it somewhere where LaTeX can find it (usually in $TEXMF/tex/latex/leipzig/). To use the leipzig package, declare it in the preamble of your document:

(1)
```
\documentclass{article}
...
\usepackage{leipzig}
```

2.2 With glossaries

If you want to use leipzig with indexing capabilities, then you will also need to download the glossaries package, v3.02 (2012/05/21) or later. Version 3.02 comes bundled with the glossary-inline package, which is necessary to make a list of abbreviations in a footnote, as is common in linguistics articles. Save it somewhere LaTeX can find it. Run `latex glossaries.ins` to generate the style files, if need be, and refer to the installation instructions in the glossaries package.

The glossaries package requires v2.5f (2006/11/18) or later of the xkeyval package. This may be a newer version than the version which came bundled with your distribution, so you should download the newest version of xkeyval from CTAN too.

Once you have both packages installed, you can use the minimal example file `minimalgls.tex` to test that glossaries is functioning correctly. See chapter 1 of the glossaries package for more information.

```
(pdf)latex minimalgls
makeglossaries minimalgls
(pdf)latex minimalgls
(pdf)latex minimalgls
```

The leipzig package should be loaded after glossaries. The glossaries package comes pre-loaded with many glossary styles, but the inline style must be loaded as a separate package.

(2)
```
\documentclass{article}
...
\usepackage{glossaries}
\usepackage{glossary-inline}
\usepackage{leipzig}
\makeglossaries
```

The glossaries package is fairly heavyweight, so some useful package options include the following (see section 2.1 of the glossaries user documentation for other package options):

(3)
```
\usepackage[nomain,nostyles]{glossaries}
```

- **nomain** - suppresses creation of a main glossary (useful if you are not using a main glossary and are only using the glossaries package to index abbreviations.)
- **nostyles** - prevents the predefined glossary styles from being loaded. If you use this package option then you must load any style you want to use manually with `\usepackage{`⟨*glossary-style*⟩`}`. The `glossary-inline` style is not automatically loaded by glossaries and must be loaded manually in any case.

2.3 With hyperref

Note that if you have also loaded the hyperref package, then glossaries must be loaded *after* hyperref, contrary to common usage. When hyperref is loaded, each usage of an abbreviation will link to the corresponding glossary entry. If you are using an inline glossary, these hyperlinks will not be very interesting, since they all link to the same page. You can turn them off with the glossaries command `\glsdisablehyper` (but note this will suppress hyperlinks for *all* acronyms and glossary entries, if you have more than one glossary). A minimal preamble declaration would then look like this:

(4)
```
\documentclass{article}
...
```

```
\usepackage{hyperref}
\usepackage{glossaries}
\usepackage{glossary-inline}
\usepackage{leipzig}
\makeglossaries
\glsdisablehyper
```

3 Package options

glossaries Using glossary indexing capabilities. Defaults to true when leipzig is loaded after glossaries.

noglossaries Turns off glossary indexing capabilities, even when leipzig is loaded after glossaries.

nostandards Use this option if you do not want leipzig to print the set of glosses defined in the Leipzig Glossing Rules. Instead it will only print non-standard ones that you define in the preamble with `\newleipzig`.

4 Usage

4.1 Abbreviation macros

The abbreviation macros are usually equivalent to the abbrevation itself, so that `\Cop{}` will typeset COP, and `\Ins{}` will typeset INS, etc. Note that the macros all begin with an uppercase letter. This makes them easier to see in your .tex file, and uppercase macros are less likely to be defined than lowercase ones. These abbreviation macros take no arguments and will gobble a following space, so they require braces. You can type either `\Cop{}` or `{\Cop}`.

There are a few notable exceptions, where the macro is not equivalent to the abbreviation, because the expected macro was already defined in LaTeX. These are shown in Table (1). Additionally, since macro names cannot start with a number, `\First{}` is the abbreviation macro for 1, `\Second{}` is the abbreviation macro for 2, and `\Third{}` is the abbreviation macro for 3.

Command	Short	Long
`\Aarg{}`	A	agent
`\Parg{}`	P	patient
`\Sarg{}`	S	argument of intransitive verb
`\First{}`	1	first person
`\Second{}`	2	second person
`\Third{}`	3	third person

Table 1: Unexpected macro names

The package also defines macros for common person-number combinations, like 1SG and 3PL. Abbreviations for first person begin with 'F' (not '1'), abbreviations for second person begin with 'S' (not '2'), and abbreviations for third person begin with 'T' (not '3'). These abbreviations are shown in Table (2).

`\Fsg{}`	1SG	`\Ssg{}`	2SG	`\Tsg{}`	3SG
`\Fdu{}`	1DU	`\Sdu{}`	2DU	`\Tdu{}`	3DU
`\Fpl{}`	1PL	`\Spl{}`	2PL	`\Tpl{}`	3PL

Table 2: Abbreviations for persons and number

4.2 Create new abbreviations

`\newleipzig` Create new abbreviations with the `\newleipzig` command, which requires three arguments.

(5) `\newleipzig [⟨options⟩] {⟨macro⟩} {⟨short⟩} {⟨long⟩}`

The optional argument ⟨*options*⟩ is a key=value list which is passed to glossaries if it is loaded. A list of recognized keys is in chapter 4 of the glossaries documentation.

The first argument ⟨*macro*⟩ should be the macro name with no backslash in front. The package will capitalize the first letter of ⟨*macro*⟩ and use the result as the macroname. The second argument ⟨*short*⟩ is the short abbreviation. This needs to be lowercase so that

`\textsc{}` will work. (You cannot make capital letters into smallcaps with `\textsc{}`.) The third argument ⟨*long*⟩ is the long version of the acronym. I also recommend typing this argument in lowercase, and using the glossaries package to format the glossary such that all long forms are consistently uppercase or lowercase. This code:

(6) `\newleipzig{vblz}{vb}{verbalizer}`

creates a macro `\Vblz{}` which will typeset VB when used.

Fusional gloss abbreviations For gloss abbreviations that are fusional, or combinations of several different grammatical glosses, you should use `\newleipzig` for each individual part. As an example, 1DU.INC is a combination of abbreviations for first person, dual number, and inclusivity. Abbreviations for first person (`\First{}`) and dual number (`\Du{}`) are already defined in `leipzig.tex`, so INC is the only part which still needs to be defined. It is generally useful to then create a shortcut macro (with `\newcommand`) for the fusional gloss which will call the abbreviations of the various parts. I usually use a command name that mirrors the abbreviations by beginning with a capital letter.

(7)
```
\newleipzig{inc}{inc}{inclusive}
\newcommand{\Fdui}{{\Fdu}.{\Inc}}%
```
. . . the `\Fdui{}` morpheme. . . % Prints 1DU.INC.

This is actually how the person-number combinations like `\Tpl{}` are defined in leipzig. The reason is so that the glossary will contain the individual components (1 = first person, SG = singular) instead of all of the various combinations thereof.

4.3 Redefine existing abbreviations

4.4 Display parameters

Gloss display The command `\newleipzig` not only creates a new definition entry, but also creates a mnemonic macro like `\Vblz{}`, which simply expands to `\gls{vblz}`. You can still access the short, long, or full forms of the abbreviation without affecting the first use flag using macros like `\acrshort`, `\Acrlong`, or `\ACRfull` (see chapter 13 of the glossaries documentation for a list). A few examples are shown in table (3).

Command	Prints
`\acrshort{vblz}`	vb
`\Acrshort{vblz}`	Vb
`\ACRshort{vblz}`	VB
`\acrlong{vblz}`	verbalizer
`\Acrlong{vblz}`	Verbalizer
`\ACRlong{vblz}`	VERBALIZER
`\acrfull{vblz}`	verbalizer (vb)
`\Acrfull{vblz}`	Verbalizer (vb)
`\ACRfull{vblz}`	VERBALIZER (VB)

Table 3: Short, long, and full formats

`\acrfullformat` The format of `\acrfull` defaults to *long-form* (*short*). You can change this by redefining `\acrfullformat` in your preamble (argument #1 is the long form and argument #2 is the short form):

(8)
```
\renewcommand{\acrfullformat}[2]{#2\space(#1)}
```

changes the order so that `\acrfull{vblz}` prints: vb (verbalizer).

5 Printing the glossary

leipzig piggybacks on the glossaries package, which allows multiple glossaries and lists of abbreviations with indexing capabilities. The command `\newleipzig` secretly calls `\newacronym` of the glossaries package and loads the acronym into the glossary type called by `\leipzigtype` (which defaults to `leipzig`, a glossary which is predefined by leipzig).

5.1 Inline glossaries

`\printglossary` Print the glossary using `\printglossary` anywhere in the document, but usually in the first footnote. You must specify the glossary style and type:

(9) `\footnote{\printglossary[style=inline,type=\leipzigtype]}`

To build the glossary, you need to LaTeX the document once, so that glossaries can index all abbreviations used and write them to an external file. Then run `makeglossaries` to build the glossary, and LaTeX twice more to print it and format it. See chapter 1 of the glossaries documentation for more information.

```
(pdf)latex myfile.tex
makeglossaries myfile
(pdf)latex myfile.tex
(pdf)latex myfile.tex
```

`glsinlineshortlongseparator` `\glsinlineseparator` `\glspostinline` See the glossary in the footnote on the first page of this documentation for the default settings of the glossary. Short and long forms of abbreviations are separated by an equals sign, glossary entries are separated by a comma, and the glossary ends in a period. You can change them by using renewcommand sometime before `\printglossaries`:

(10)
```
\renewcommand{\glsinlineseparator}{,\space}
\renewcommand{\glsinlineshortlongseparator}{$\sim$}
\renewcommand{\glspostinline}{}
```

Please note that the command `\glsinlineshortlongseparator` is not defined in the glossaries package, but only by the leipzig package. If the `glossary-inline` style is loaded, then leipzig redefines `theglossary` environment, where the glossary is printed, to include this functionality.

`\glsnamefont` The abbreviations within the glossary are formatted with `\glsnamefont`; leipzig uses smallcaps by default.

(11) `\renewcommand{\glsnamefont}[1]{\textbf{#1}}%`

`\leipzigname` `\glossarysection` The name of the gloss abbreviations glossary defaults to 'Abbreviations' and is printed as the first argument in `\glossarysection`, which defaults to print nothing in leipzig. The name is controlled by `\leipzigname`, so to change the name you can redefine `\leipzigname`:

(12)
```
\renewcommand{\leipzigname}{My new glossary header}
\renewcommand*{\glossarysection}[2][]{\textit{#1}:\space}%
```

Then I will re-print the glossary in this footnote[2], using the parameters redefined above. Compare this glossary to that on the first page to see the differences.

Here is the code I used to print that footnote:

(13)
```
...in this footnote\footnote{\printglossary[type=\leipzigtype,
style=inline]does not end in a period, to illustrate a
change from the default}, using the...
```

5.2 Block glossaries

If you are writing a book or lengthy manuscript, you probably want the glossary to be printed on a separate page in the frontmatter. The glossaries package comes pre-packaged with several different glossary styles, based on tabular environments, list environments, and more. You can also custom-build a glossary style. Below is one which I have been using. The code is in section (B).

[2] *My new glossary header*: **1**∼first person , **2**∼second person , **3**∼third person, **acc**∼accusative, **cop**∼copula, **dat**∼dative , **du**∼dual, **inc**∼inclusive , **ins**∼instrumental, **nom**∼nominative, **pl**∼plural, **sg**∼singular, **vb**∼verbalizer does not end in a period, to illustrate a change from the default

6 Abbreviations

1	First person
2	Second person
3	Third person
ACC	Accusative
COP	Copula
DAT	Dative
DU	Dual
INC	Inclusive
INS	Instrumental
NOM	Nominative
PL	Plural
SG	Singular
VB	Verbalizer

7 Multiple lists of acronyms

What do you do if your paper needs not only a list of gloss abbreviations, but also some other list(s) of acronyms, possibly formatted differently than the gloss abbreviations? All abbreviations declared with `\newleipzig` are saved in the `leipzig` glossary. This means that you can still use the `acronym` function in the glossaries package for other acronyms (or define your own glossary). Use the `acronym` package option:

(14)
```
\usepackage[acronym]{glossaries}
```

and declare new acronyms with `\newacronym`:

(15)
```
\newacronym[⟨options⟩]{⟨unique-id⟩}{⟨short⟩}{⟨long⟩}
```

and refer to chapter 13 in the glossaries user documentation for more information about how to create acronym lists.

The leipzig package defines a newglossary called `leipzig`:

(16) `\newglossary{leipzig}{lzs}{lzo}{\leipzigname}`

Gloss abbreviations defined with `\newleipzig` are added to glossary type `\leipzigtype`, which defaults to `leipzig`. (This is so that if you have other lists of acronyms in your paper, they will not be formatted the same as the gloss abbreviations.) You can redefine `\leipzigtype` to some other glossary if you want abbreviations to be added to some other glossary of your choice. The following line of code will instead put all gloss abbreviations into `\acronymtype`, which is a macro defined by the glossaries package. If the `acronym` package option was declared, then `\acronymtype` points to the `acronym` glossary (pre-defined in glossaries); if not, then `\acronymtype` points to the `main` glossary (also pre-defined by glossaries).

(17) `\renewcommand{\leipzigtype}{\acronymtype}`

8 Known bugs

Using leipzig for glossing abbreviations alongside other glossaries and/or lists of acronyms is largely untested but probably has unexpected results. Please contact me with issues.

References

[1] Bickel, Balthasar, Bernard Comrie, and Martin Haspelmath. (2008). "The Leipzig Glossing Rules. Conventions for Interlinear Morpheme by Morpheme Glosses." Revised version of February 2008. Department of Linguistics, Max Plank Institute for Evolutionary Anthropology. Retreived 30 June 2012: http://www.eva.mpg.de/lingua/resources/glossing-rules.php.

A Pre-defined abbreviations

Command	Short	Long
`\First{}`	1	first person
`\Second{}`	2	second person
`\Third{}`	3	third person
`\Abl{}`	ABL	ablative
`\Abs{}`	ABS	absolutive
`\Acc{}`	ACC	accusative
`\Adj{}`	ADJ	adjective
`\Adv{}`	ADV	adverbial
`\Aarg{}`	A	agent
`\Agr{}`	AGR	agreement
`\All{}`	ALL	allative
`\Antip{}`	ANTIP	antipassive
`\Appl{}`	APPL	applicative
`\Art{}`	ART	article
`\Aux{}`	AUX	auxiliary
`\Ben{}`	BEN	benefactive
`\Caus{}`	CAUS	causative
`\Clf{}`	CLF	classifier
`\Com{}`	COM	comitative
`\Comp{}`	COMP	complementizer
`\Compl{}`	COMPL	completive
`\Cond{}`	COND	conditional
`\Cop{}`	COP	copula
`\Cvb{}`	CVB	converb
`\Dat{}`	DAT	dative
`\Decl{}`	DECL	declarative
`\Defn{}`	DEF	definite
`\Dem{}`	DEM	demonstrative
`\Det{}`	DET	determiner
`\Dist{}`	DIST	distal
`\Distr{}`	DISTR	distributive
`\Du{}`	DU	dual
`\Dur{}`	DUR	durative
`\Erg{}`	ERG	ergative

`\Excl{}`	EXCL	exclusive
`\F{}`	F	feminine
`\Foc{}`	FOC	focus
`\Fut{}`	FUT	future
`\Gen{}`	GEN	genitive
`\Imp{}`	IMP	imperative
`\Incl{}`	INCL	inclusive
`\Ind{}`	IND	indicative
`\Indf{}`	INDF	indefinite
`\Inf{}`	INF	infinitive
`\Ins{}`	INS	instrumental
`\Intr{}`	INTR	intransitive
`\Impf{}`	IMPF	imperfective
`\Irr{}`	IRR	irrealis
`\Loc{}`	LOC	locative
`\M{}`	M	masculine
`\N{}`	N	neuter
`\Neg{}`	NEG	negative
`\Nmlz{}`	NMLZ	nominalizer
`\Nom{}`	NOM	nominative
`\Obj{}`	OBJ	object
`\Obl{}`	OBL	oblique
`\Pass{}`	PASS	passive
`\Parg{}`	P	patient
`\Pfv{}`	PFV	perfective
`\Pl{}`	PL	plural
`\Poss{}`	POSS	possessive
`\Pred{}`	PRED	predicative
`\Prf{}`	PRF	perfect
`\Prs{}`	PRS	present
`\Prog{}`	PROG	progressive
`\Proh{}`	PROH	prohibitive
`\Prox{}`	PROX	proximal
`\Pst{}`	PST	past
`\Ptcp{}`	PTCP	participle
`\Purp{}`	PURP	purposive
`\Q{}`	Q	question particle
`\Quot{}`	QUOT	quotative

`\Recp{}`	RECP	reciprocal
`\Refl{}`	REFL	reflexive
`\Rel{}`	REL	relative
`\Res{}`	RES	resultative
`\Sbj{}`	SBJ	subject
`\Subj{}`	SUBJ	subjunctive
`\Sg{}`	SG	singular
`\Sarg{}`	S	argument of intransitive argument
`\Top{}`	TOP	topic
`\Tr{}`	TR	transitive
`\Voc{}`	VOC	vocative

B Custom glossary environment

```
\documentclass{book}

 \usepackage[nomain,section=chapter]{glossaries}%
    [2012/05/21 v3.02 (NLCT)]%
 \usepackage{glossary-inline}%
 \newglossarystyle{mysuper}{%
 \glossarystyle{super}% based on super
   \renewenvironment{theglossary}%
     {\tablehead{}\tabletail{}%
      \begin{supertabular}{@{}lp{\glsdescwidth}}}%
     {\end{supertabular}}%
   \renewcommand*{\glossaryheader}{}%
   \renewcommand*{\glsgroupheading}[1]{}%
   \renewcommand*{\glossaryentryfield}[5]{%
     \glsentryitem{##1}\glstarget{##1}{##2}
       & \makefirstuc{##3}\glspostdescription{}\\}%
   \renewcommand*{\glossarysubentryfield}[6]{%
      &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\makefirstuc{##4}\glspostdescription{}\\}%
   \renewcommand*{\glsgroupskip}{}%
 }%
 \usepackage{leipzig}%
 \makeglossaries
 \glsdisablehyper
```

```
\begin{document}
\printglossary[style=mysuper,type=\leipzigtype]
\end{document}
```

C The Code

This code originally Sven Siegmund's, created with help of XeTeX mailing list, to detect optional argument. See http://xelatex.blogspot.com/2008/03/newcommand-with-optional-argument.html.

```
\long\def\tlist@if@empty@nTF #1{%
  \expandafter\ifx\expandafter\\\detokenize{#1}\\%
  \expandafter\@firstoftwo%
  \else%
  \expandafter\@secondoftwo%
  \fi%
}%
```

Some booleans to determine whether the glossaries package is loaded or not. The idea was to be able to let users load the leipzig class either before or after glossaries, but they do not currently work well.

```
\newif\ifleipzig@glossaries\leipzig@glossariesfalse
\newif\ifleipzig@noglossaries\leipzig@noglossariesfalse
```

A boolean to not index the abbreviations defined in the Leipzig Glossing Rules.

```
\newif\ifleipzig@donotindex\leipzig@donotindexfalse
```

`glossaries` `noglossaries` Users can specify with package options whether they would like to use leipzig with the indexing capabilities of glossaries. For instance, if the glossaries package is loaded but the user does not want to index abbreviations, then he can use the `noglossaries` package option.

```
\DeclareOption{glossaries}{\leipzig@glossariestrue}
\DeclareOption{noglossaries}{\leipzig@noglossariestrue}
%\RequirePackage{xkeyval}
%\newlength{\preview@border}
%\setlength{\preview@border}{0pt}
%\DeclareOptionX{PreviewBorder}[0pt]{\setlength{\preview@border}{#1}
%\ProcessOptionsX
```

nostandards Use this package option if you do not want the standard Leipzig abbreviations to show up in the glossary.

```
\DeclareOption{nostandards}{\leipzig@donotindextrue}
```

Pass any other options to the glossaries package and process options.

```
\DeclareOption*{%
  \PassOptionsToPackage{\CurrentOption}{glossaries}%
}%
\ProcessOptions\relax
```

Determine if the glossaries package was loaded, and set the boolean to true (unless overwritten by the user with the package option `noglossaries`).

```
\@ifpackageloaded{glossaries}{% true text
  \ifleipzig@noglossaries\leipzig@glossariesfalse\relax
  \else\leipzig@glossariestrue\relax\fi
}{% false text
  \ifleipzig@glossaries\PackageWarning{leipzig}{%
    glossaries package not loaded. Load glossaries
    package at some point.}%
  \else\leipzig@glossariesfalse\relax\fi
}%
```

If glossary-inline style not loaded, do nothing. Else renew inline style to have a user-defined separator between the abbreviation and long form.

```
\@ifpackageloaded{glossary-inline}{% renew inline
\renewglossarystyle{inline}{%
  \renewenvironment{theglossary}%
    {%
      \def\gls@inlinesep{}%
      \def\gls@inlinesubsep{}%
      \def\gls@inlineshortlongsep{%
          \glsinlineshortlongseparator}% added this
    }%
    {\glspostinline}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand*{\glossaryentryfield}[5]{%
    \gls@inlinesep
    \glsentryitem{##1}\glstarget{##1}{##2}%
    \def\glo@desc{##3}%
    \def\@no@post@desc{\nopostdesc}%
    \ifx\glo@desc\@no@post@desc
    \else
      \ifstrempty{##3}{}{\gls@inlineshortlongsep##3}%
    \fi
    \ifglshaschildren{##1}%
    {%
       \glsresetsubentrycounter
       \glsinlineparentchildseparator
       \def\gls@inlinesubsep{}%
    }%
    {}%
    \def\gls@inlinesep{\glsinlineseparator}%
  }%
  \renewcommand*{\glossarysubentryfield}[6]{%
    \gls@inlinesubsep%
    \glstarget{##2}{}%
    \glssubentryitem{##2}##4%
    \def\gls@inlinesubsep{\glsinlinesubseparator}%
  }%
  \renewcommand*{\glsgroupskip}{}%
}
```

`ineshortlongseparator` `glsnamefont` `glossarysection` The separator between the short and long forms in the glossary can be set to a user-defined style like {=} or `{:\space}` (default is `\space`). By default the short form of the abbreviations are set to smallcaps, and there is no glossary section title.

```
\newcommand*{\glsinlineshortlongseparator}{\,=\,\linebreak[1]}% added this
\renewcommand*{\glsinlineseparator}{,\space}% changed from ;\space
\renewcommand*{\glsinlinesubseparator}{,\space}
\renewcommand*{\glsinlineparentchildseparator}{:\space}
\renewcommand*{\glspostinline}{.\space}% changed from .
\renewcommand{\glsnamefont}[1]{\textsc{#1}}% abbrv in glossary are \sc
\renewcommand*{\glossarysection}[2][]{}% no section name
}{\relax}% if glossary-inline not loaded
```

```
% Default is to print all grammatical glosses in small caps:
\newcommand{\leipzigfont}[1]{\textsc{#1}}%
\newcommand{\firstleipzigfont}[1]{\leipzigfont{#1}}%
```

```
\ifleipzig@glossaries %if glossaries package loaded
  \renewcommand*{\acrpluralsuffix}{\textup{\glspluralsuffix}}%
  \newcommand{\leipzigname}{Abbreviations}
  \newglossary{leipzig}{lzs}{lzo}{\leipzigname}
  \newcommand{\leipzigtype}{leipzig}
%% The following commands are based on the Custom Acronym commands in the glossaries
% %% #1 = first / firstplural / text / plural (as appropriate
% %% #2 = description
% %% #3 = symbol
% %% #4 = inserted text
\newcommand*{\SetLeipzigDisplayStyle}[1]{%
  \defglsdisplay[#1]{\leipzigfont{##1}##4}%
  \defglsdisplayfirst[#1]{\firstleipzigfont{##1}##4}%
}
\newcommand*{\CustomLeipzigFields}{%
  name={\the\glsshorttok},%
  description={\the\glslongtok},%
  symbol={\the\glsshorttok},%
  first={\firstleipzigfont{\the\glsshorttok}},%
  firstplural={\firstleipzigfont{\the\glsshorttok}\noexpand\acrpluralsuffix},%
  text={\leipzigfont{\the\glsshorttok}},%
  plural={\leipzigfont{\the\glsshorttok}\noexpand\acrpluralsuffix}%
}
\newcommand*{\LeipzigAcronymDef}{%
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\leipzigtype,%
      short={\leipzigfont{\the\glsshorttok}},% used in e.g. \acrshort
      shortplural={\leipzigfont{\the\glsshorttok}\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      user1={\the\glsshorttok},%
      user2={\the\glsshorttok\noexpand\acrpluralsuffix},%
      user3={\the\glslongtok},%
      user4={\the\glslongtok\noexpand\acrpluralsuffix},%
      \CustomLeipzigFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}
\newcommand*{\SetLeipzigStyle}{%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@glsacronymlists\@empty
      \def\@glo@type{\leipzigtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetLeipzigDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \LeipzigAcronymDef
  }%
%%  \@for\@gls@type:=\@glsacronymlists\do{% sets style for all acronymlists?
%%%%    \SetCustomDisplayStyle{\@gls@type}%
%%    \SetLeipzigDisplayStyle{\@gls@type}%
%%  }%
}%
\SetLeipzigStyle
%%Normally would have |\leipzigfont{#2}|, but I coded the leipzigfont directly into
\renewcommand{\acrfullformat}[2]{#1\space(#2)}
\newcommand{\newleipzig}[4][]{%
  \bgroup
    \tlist@if@empty@nTF{#1}%
      {\newacronym[type=\leipzigtype]{#2}{#3}{#4}}%
      {\newacronym[type=\leipzigtype,#1]{#2}{#3}{#4}}%
 %   \newacronym[type=\leipzigtype][#1]{n#2}{n#3}{Non-#4}
    \@newleipzig(#3)#2\@nil}%
  \def\@newleipzig(#1)#2#3\@nil{%
```

If the `leipzig@donotindex` boolean is toggled, then `\gls` will not be called whenever the macro abbreviation is called. That way, these abbreviations are never indexed and won't show up in the glossary, but they will be formatted just like other abbreviations.

```
      \ifleipzig@donotindex
        \uppercase{\expandafter\gdef\csname #2}#3\endcsname{\leipzigfont{#1}}
```

Otherwise, `\gls` is called, and any time the macro abbreviation is used, this acronym will be indexed and appear in the glossary.

```
      \else
        \uppercase{\expandafter\gdef\csname #2}#3\endcsname{\gls{#2#3}}
      \fi
      \egroup
    }% end if glossaries loaded
```

If glossaries not loaded, then the code is much shorter:

```
\else % if glossaries not loaded
  \newcommand{\newleipzig}[4][]{\@newleipzig(#3)#2\@nil}%
  \def\@newleipzig(#1)#2#3\@nil{%
    \uppercase{\expandafter\gdef\csname #2}#3\endcsname{\leipzigfont{#1}}
  }%
\fi
```

Finally, load the standard gloss abbreviations.

```
\input{leipzig.tex}
\ifleipzig@donotindex\leipzig@donotindexfalse\fi
```