

Documented Code For glossaries v4.56

Nicola L.C. Talbot

Dickimaw Books

<http://www.dickimaw-books.com/>

2025-03-03

Contents

1	Introduction	3
2	Main Package Code (glossaries.sty)	4
2.1	Package Definition	4
2.2	Package Options	7
2.3	Predefined Text	47
2.4	Xindy	57
2.5	Loops and conditionals	66
2.6	Defining new glossaries	74
2.7	Defining new entries	78
2.8	Resetting and unsetting entry flags	106
2.9	Keeping Track of How Many Times an Entry Has Been Unset . .	111
2.10	Loading files containing glossary entries	116
2.11	Using glossary entries in the text	117
	2.11.1 Links to glossary entries	122
	2.11.2 Displaying entry details without adding information to the glossary	169
2.12	Adding an entry to the glossary without generating text	180
2.13	Creating associated files	182
2.14	Writing information to associated files	204
2.15	Glossary Entry Cross-References	214
2.16	Displaying the glossary	216
2.17	Acronyms	250
2.18	Predefined acronym styles	255
2.19	Predefined Glossary Styles	273
2.20	Debugging Commands	274
2.21	Compatibility with version 2.07 and below	279

3	Glossary Styles	280
3.1	Glossary hyper-navigation definitions (<code>glossary-hypernav.sty</code>) . . .	280
3.2	In-line Style (<code>glossary-inline.sty</code>)	284
3.3	List Style (<code>glossary-list.sty</code>)	286
3.4	Glossary Styles using <code>longtable</code> (<code>glossary-long.sty</code>)	290
3.5	Glossary Styles using <code>longtable</code> and <code>booktabs</code> (<code>glossary-longbooktabs.sty</code>)	297
3.6	Glossary Styles using <code>longtable</code> (<code>glossary-longragged.sty</code>)	302
3.7	Glossary Styles using <code>multicol</code> (<code>glossary-mcols.sty</code>)	307
3.8	Glossary Styles using <code>supertabular</code> environment (<code>glossary-super.sty</code>)	313
3.9	Glossary Styles using <code>supertabular</code> environment (<code>glossary-superragged.sty</code>)	320
3.10	Tree Styles (<code>glossary-tree.sty</code>)	327
4	Multi-Lingual Support	336
4.1	Babel Captions (<code>glossaries-babel.sty</code>)	336
4.2	Polyglossia Captions (<code>glossaries-polyglossia.sty</code>)	337
5	Prefix Support (<code>glossaries-prefix Code</code>)	338
6	Accessibility Support (<code>glossaries-accsupp Code</code>)	345
6.1	Defining Replacement Text	347
6.2	Accessing Replacement Text	351
6.3	Displaying the Glossary	371
6.4	Acronyms	373
6.5	Debugging Commands	383
7	Backwards Compatibility	384
7.1	<code>glossaries-compatible-207</code>	384
7.2	<code>glossaries-compatible-307</code>	384
8	Rollback	385
8.1	Rollback v4.46 (<code>glossaries-2020-03-19.sty</code>)	385
8.2	Rollback v4.46 (<code>glossaries-accsupp-2020-03-19.sty</code>)	548
8.3	Rollback v4.46 (<code>glossaries-babel-2020-03-19.sty</code>)	583
8.4	Rollback v4.46 (<code>glossaries-compatible-207-2020-03-19.sty</code>)	584
8.5	Rollback v4.46 (<code>glossaries-compatible-307-2020-03-19.sty</code>)	589
8.6	Rollback v4.46 (<code>glossaries-polyglossia-2020-03-19.sty</code>)	600
8.7	Rollback v4.46 (<code>glossaries-prefix-2020-03-19.sty</code>)	600
8.8	Rollback v4.46 (<code>glossary-hypernav-2020-03-19.sty</code>)	605
8.9	Rollback v4.46 (<code>glossary-inline-2020-03-19.sty</code>)	606
8.10	Rollback v4.46 (<code>glossary-list-2020-03-19.sty</code>)	607
8.11	Rollback v4.46 (<code>glossary-list-2020-03-19.sty</code>)	609
8.12	Rollback v4.46 (<code>glossary-longbooktabs-2020-03-19.sty</code>)	612
8.13	Rollback v4.46 (<code>glossary-longragged-2020-03-19.sty</code>)	615
8.14	Rollback v4.46 (<code>glossary-mcols-2020-03-19.sty</code>)	618
8.15	Rollback v4.46 (<code>glossary-super-2020-03-19.sty</code>)	622
8.16	Rollback v4.46 (<code>glossary-superragged-2020-03-19.sty</code>)	626

8.17	Rollback v4.46 (glossary-tree-2020-03-19.sty)	629
8.18	Rollback v4.49 (glossaries-2021-11-01.sty)	635
8.19	Rollback v4.49 (glossaries-accsupp-2021-11-01.sty)	801
8.20	Rollback v4.49 (glossaries-babel-2021-11-01.sty)	836
8.21	Rollback v4.49 (glossaries-compatible-207-2021-11-01.sty)	837
8.22	Rollback v4.49 (glossaries-compatible-307-2021-11-01.sty)	843
8.23	Rollback v4.49 (glossaries-polyglossia-2021-11-01.sty)	853
8.24	Rollback v4.49 (glossaries-prefix-2021-11-01.sty)	853
8.25	Rollback v4.49 (glossary-hypernav-2021-11-01.sty)	858
8.26	Rollback v4.49 (glossary-inline-2021-11-01.sty)	859
8.27	Rollback v4.49 (glossary-list-2021-11-01.sty)	861
8.28	Rollback v4.49 (glossary-long-2021-11-01.sty)	863
8.29	Rollback v4.49 (glossary-longbooktabs-2021-11-01.sty)	866
8.30	Rollback v4.49 (glossary-longragged-2021-11-01.sty)	869
8.31	Rollback v4.49 (glossary-mcols-2021-11-01.sty)	872
8.32	Rollback v4.49 (glossary-super-2021-11-01.sty)	875
8.33	Rollback v4.49 (glossary-superragged-2021-11-01.sty)	879
8.34	Rollback v4.49 (glossary-tree-2021-11-01.sty)	883
8.35	Rollback v4.52 (glossaries-2022-11-03.sty)	888
8.36	Rollback v4.52 (glossary-hypernav-2022-11-03.sty)	1042
9	Rollback	1043
9.1	Rollback v4.54 (glossaries-2024-04-03.sty)	1043

1 Introduction

This is the documented code for the `glossaries` package. This bundle comes with the following documentation:

`glossariesbegin.pdf` If you are a complete beginner, start with “The glossaries package: a guide for beginners”.

`glossary2glossaries.pdf` If you are moving over from the obsolete `glossary` package, read “Upgrading from the `glossary` package to the `glossaries` package”.

`glossaries-user.pdf` For the main user guide, read “`glossaries.sty` v4.56: L^AT_EX₂ ϵ Package to Assist Generating Glossaries”.

`mfirstuc-manual.pdf` The commands provided by the `mfirstuc` package are briefly described in “`mfirstuc.sty`: uppercasing first letter”.

`glossaries-code.pdf` This document is for advanced users wishing to know more about the inner workings of the `glossaries` package.

`INSTALL` Installation instructions.

`CHANGES` Change log.

README.md Package summary.

The user level commands described in the user manual (glossaries-user.pdf) may be considered “future-proof”. Even if they become deprecated, they should still work for old documents (although they may not work in a document that also contains new commands introduced since the old commands were deprecated, and you may need to specify a compatibility mode).

The internal commands in *this* document that aren’t documented in the *user manual* should not be considered future-proof and are liable to change. If you want a new user level command, you can post a feature request at <http://www.dickimaw-books.com/feature-request.html>. If you are a package writer wanting to integrate your package with `glossaries`, it’s better to request a new user level command than to hack these internals.

2 Main Package Code (glossaries.sty)

2.1 Package Definition

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossaries-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-2021-11-01.sty}
\DeclareRelease{v4.52}{2022-11-03}{glossaries-2022-11-03.sty}
\DeclareRelease{v4.54}{2024-04-03}{glossaries-2024-04-03.sty}
\DeclareCurrentRelease{v4.56}{2025-03-03}
```

Declare package:

```
\ProvidesPackage{glossaries}[2025/03/03 v4.56 (NLCT)]
```

Required packages:

```
\RequirePackage{ifthen}
\RequirePackage{xkeyval}[2006/11/18]
\RequirePackage{mfirstuc}
```

`\glsentencecase` Provided to allow user to switch to `\MFUsentencecase` if preferred. Don’t redefine `\glsmakefirstuc` to `\makefirstuc` as it will result in infinite recursion.

```
\newcommand{\glsentencecase}[1]{\makefirstuc{#1}}
```

```
\ExplSyntaxOn
```

`\glslowercase` Provided to make it easier for user to use \LaTeX 3 lowercase command without switching to `exp3` syntax. For example, to redefine `\acronymfont` to automatically convert to lowercase.

```
\newcommand{\glslowercase}[1]{\text_lowercase:n { #1 } }
```

`\glsuppercase` Note that this isn’t using `\mfirstucMakeUppercase`, which will be defined to use the robust `\MakeUppercase` if an old version of `mfirstuc` is installed.

```
\newcommand{\glsuppercase}[1]{\text_uppercase:n { #1 } }
```

Check if we have `mfirstuc v2.08+`. In which case, don't redefine `\mfirstucMakeUppercase`. Since `\MFUsentencecase` was new in v2.08, the simplest check is to test for the existence of that command.

```
\ifdef\MFUsentencecase
{%
```

Automatically identify exclusions, blockers and mappings.

```
\glsmfuexcl
```

```
\newcommand{\glsmfuexcl}[1]{\MFUexcl{#1}}
```

```
\glsmfublocker
```

```
\newcommand{\glsmfublocker}[1]{\MFUblocker{#1}}
```

```
\glsmfuaddmap
```

```
\newcommand{\glsmfuaddmap}[2]{\MFUaddmap{#1}{#2}}
```

```
}
```

```
{
```

The `textcase` package is now deprecated, so only load it with old versions of `mfirstuc`. (From 2022, `textcase` simply defines `\MakeTextUppercase` to `\MakeUppercase`.)

```
\RequirePackage{textcase}
```

Provide `\MFUsentencecase` for use where expandable contexts are required.

```
\MFUsentencecase
```

```
\providecommand{\MFUsentencecase}[1]{\text_titlecase_first:n { #1 } }
```

Provide support for exclusions with `\MFUsentencecase`.

```
\glsmfuexcl
```

```
\providecommand{\glsmfuexcl}[1]{
\text_titlecase_exclude_arg_t1 {#1}
{
\text_titlecase_exclude_arg_t1 {#1}
}
}
```

`\mfirstucMakeUppercase` We have an old version of `mfirstuc` that is still using this command in the definition of `\makefirstuc`, in which case, continue to use `\MakeTextUppercase` as in previous versions, in case there is also an old version of `textcase` installed. However, note that the all-caps commands like `\GLS` now use `\glssupercase` instead.

```
\renewcommand*{\mfirstucMakeUppercase}{\MakeTextUppercase}
```

Just treat blockers and mappings as exclusions.

```
\glsmfublocker
```

```
\newcommand{\glsmfublocker}[1]{\glsmfuexcl{#1}}
```

```

\glsmfuaddmap
    \newcommand{\glsmfuaddmap}[2]{\glsmfuexcl{#1}\glsmfublocker{#2}}
}
\ExplSyntaxOff
\RequirePackage{xfor}

```

Need to use `\new@ifnextchar` instead of `\@ifnextchar` in commands that have a final optional argument (such as `\gls`) so require `amsgen`. Thanks to Morten Høgholm for suggesting this. (This has replaced using the `xspace` package.)

```
\RequirePackage{amsgen}
```

As from v3.0, now loading `etoolbox` (this is now redundant as `datatool`-base loads `etoolbox`):

```
\RequirePackage{etoolbox}
```

Check if `doc` has been loaded.

```
\if@gls@docloaded
```

```
\newif\if@gls@docloaded
```

```
\@ifpackageloaded{doc}{%
```

```
{%
```

```
\@gls@docloadedtrue
```

```
}%
```

```
{%
```

```
\@ifclassloaded{nlctdoc}{\@gls@docloadedtrue}{\@gls@docloadedfalse}%
```

```
}
```

```
\if@gls@docloaded
```

`\doc` has been loaded, so some modifications need to be made to ensure both packages can work together. The amount of conflict has been reduced as from v4.11 and no longer involves patching internal commands.

`\PrintChanges` needs to use `doc`'s version of `theglossary`, so save that.

```
\glsorg@theglossary
```

```
\let\glsorg@theglossary\theglossary
```

```
\glsorg@endtheglossary
```

```
\let\glsorg@endtheglossary\endtheglossary
```

`\PrintChanges` Now redefine `\PrintChanges` so that it uses the original `theglossary` environment.

```
\let\glsorg@PrintChanges\PrintChanges
```

```
\renewcommand{\PrintChanges}{%
```

```
\begingroup
```

```
\let\theglossary\glsorg@theglossary
```

```
\let\endtheglossary\glsorg@endtheglossary
```

```
\glsorg@PrintChanges
```

```
\endgroup
```

```
}
```

End of doc stuff.

\fi

2.2 Package Options

`debug` Switch on debug mode. This will also cancel the `nowarn` option. This is now a choice key.

```
\newif\if@gldebug
\define@choicekey{glossaries.sty}{debug}{\gldebug@val\gldebug@nr}%
{true,false,showtargets,showaccsupp}[true]{%
\ifcase\gldebug@nr\relax
% debug=true
\@gldebugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\let\glshowtarget\@gobble
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\or
% debug=false
\@gldebugfalse
\let\glshowtarget\@gobble
\let\glshowaccsupp\@gobblethree
\PackageInfo{glossaries}{debug mode OFF}%
\or
% debug=showtargets
\@gldebugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\glshowtarget}{\@glshowtarget}%
\or
% debug=showaccsupp
\@gldebugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\glshowaccsupp}{\glshowaccsupp}%
```

```

    \fi
  }

```

`\glsshowtarget` If `debug=showtargets`, show the hyperlink target name in the margin.

```

\newcommand*\glsshowtarget}[1]{%
  \ifmmode
    \glsshowtargetinner{#1}%
  \else
    \ifinner
      \glsshowtargetinner{#1}%
    \else
      \glsshowtargetouter{#1}%
    \fi
  \fi
}

```

`\glsshowtargetfonttext` Text-block command that checks for math-mode.

```

\newcommand*\glsshowtargetfonttext}[1]{%
  \ifmmode \nfss@text{\glsshowtargetfont #1}\else {\glsshowtargetfont #1}\fi
}

```

`\glsshowtargetinner` Grouping no longer required as new `\@@glsshowtarget` adds scoping but retained here in case any existing documents are using

```

\newcommand*\glsshowtargetinner}[1]{\glsshowtargetfonttext{[#1]}}

```

`\glsshowtargetouter`

```

\newcommand*\glsshowtargetouter}[1]{%
  \glsshowtargetsymbol\marginpar{\glsshowtargetsymbol\glsshowtargetfont #1}}

```

`\glsshowtargetsymbol`

```

\newcommand*\glsshowtargetsymbol>{{\tiny$\triangleright$}}

```

`\glsshowtargetfont`

```

\newcommand*\glsshowtargetfont{\ttfamily\footnotesize}

```

`\@glsshowtarget` `debug=showtargets` will redefine this.

```

\newcommand*\@glsshowtarget}[1]{

```

`\glsdoshowtarget` Make it easier to customise.

```

\newcommand*\glsdoshowtarget}[2]{\@glsshowtarget{#1}#2}

```

`\@@glsshowtarget` Need to detokenize the label in the event that it contains awkward characters like underscores.

```

\newrobustcmd*\@@glsshowtarget}[1]{%
  \begingroup
  \protected@edef\@gls@tmp{#1}%
  \@onelevel@sanitize\@gls@tmp
  \expandafter\glsshowtarget\expandafter{\@gls@tmp}%
  \endgroup
}

```


`\@glsshowaccsupp debug=showaccsupp` will redefine this.

```
\newcommand*\@glsshowaccsupp}[3]{}
```

`\glsshowaccsupp` Just use `\@glsshowtarget` since it basically needs to do the same thing.

```
\newrobustcmd*\glsshowaccsupp}[3]{%
  \ifstrempy{#1}%
  {\@glsshowtarget{/#2 (#3)}}%
  {\@glsshowtarget{/#2 (#3) [#1]}}%
}
```

Determine what to do if the `see` key is used before `\makeglossaries`. The default is to produce an error.

`\@gls@see@noindex`

```
\newcommand*\@gls@see@noindex){%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries\space (or move
  \string\newglossaryentry\space
  definitions into the preamble)}%
  {You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key. It may
  be that the 'see' key has been written to the .glsdefs
  file from the previous run, in which case you need to
  move your definitions
  to the preamble if you don't want to use
  \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
```

`seenoindex`

```
\define@choicekey{glossaries.sty}{seenoindex}%
[\gls@seenoindex@val\gls@seenoindex@nr]{error,warn,ignore}{%
  \ifcase\gls@seenoindex@nr
  \renewcommand*\@gls@see@noindex){%
    \PackageError{glossaries}%
    {'\gls@xr@key' key may only be used after \string\makeglossaries\space
    or \string\makenoidxglossaries}%
    {You must use \string\makeglossaries\space
    or \string\makenoidxglossaries\space before defining
    any entries that have a '\gls@xr@key' key}%
  }%
  \or
  \renewcommand*\@gls@see@noindex){%
    \GlossariesWarning{'\gls@xr@key' key ignored}%
  }%
  \or
  \renewcommand*\@gls@see@noindex){}%
```

```

\fi
}

```

toc The toc package option will add the glossaries to the table of contents. This is a boolean key, if the value is omitted it is taken to be true.

```
\define@boolkey{glossaries.sty}[gls]{toc}[true]{}

```

numberline The numberline package option adds `\numberline` to `\addcontentsline`. Note that this option only has an effect if used in with `toc=true`.

```
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}

```

\@@glossarysec The sectional unit used to start the glossary is stored in `\@@glossarysec`. If chapters are defined, this is initialised to `chapter`, otherwise it is initialised to `section`.

```

\ifcsundef{chapter}%
  {\newcommand*\@@glossarysec}{section}}%
  {\newcommand*\@@glossarysec}{chapter}}

```

section The section key can be used to set the sectional unit. If no unit is specified, use `section` as the default. The starred form of the named sectional unit will be used. If you want some other way to start the glossary section (e.g. a numbered section) you will have to redefined `\glossarysection`.

```

\define@choickey{glossaries.sty}{section}{part,chapter,section,%
subsection,subsubsection,paragraph,subparagraph}[section]{%
  \renewcommand*\@@glossarysec{#1}}

```

Determine whether or not to use numbered sections.

\@@glossarysecstar

```
\newcommand*\@@glossarysecstar}{*}

```

\@@glossaryseclabel

```
\newcommand*\@@glossaryseclabel}{}

```

\glsautoprefix Prefix to add before label if automatically generated:

```
\newcommand*\glsautoprefix}{}

```

numberedsection

```

\define@choickey{glossaries.sty}{numberedsection}%
  [\gls@numberedsection@val\gls@numberedsection@nr]{%
false,nolabel,autolabel,nameref}[nolabel]{%
  \ifcase\gls@numberedsection@nr\relax
    \renewcommand*\@@glossarysecstar}{*}%
    \renewcommand*\@@glossaryseclabel}{}%
  \or
    \renewcommand*\@@glossarysecstar}{}%
    \renewcommand*\@@glossaryseclabel}{}%
  \or

```

```

\renewcommand*\@@glossarysecstar}{}%
\renewcommand*\@@glossaryseclabel}{%
\label{\glsautoprefix\@glo@type}}%
\or
\renewcommand*\@@glossarysecstar}{*}%
\renewcommand*\@@glossaryseclabel}{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}}%
\fi
}

```

The default glossary style is stored in `\@glossary@default@style`. This is initialised to `list`. (The `list` style is defined in the accompanying `glossary-list` package described in [subsection 2.19](#).) Note that the `list` style is incompatible with `classicthesis` so change the default to `index` if that package has been loaded.

```

\@glossary@default@style
\@ifpackageloaded{classicthesis}
{\newcommand*\@glossary@default@style}{index}}
{\newcommand*\@glossary@default@style}{list}}

```

`style` The default glossary style can be changed using the `style` package option. The value can be the name of any defined glossary style. The glossary style is set at the beginning of the document, so you can still use the `style` key to set a style that is defined in another package. This package comes with some predefined styles that are defined in [subsection 2.19](#). This now uses `\def` instead of `\renewcommand` as `\@glossary@default@style` may have been set to `\relax`.

```

\define@key{glossaries.sty}{style}{%
\def\@glossary@default@style{#1}%
}

```

Each `\DeclareOptionX` needs a corresponding `\DeclareOption` so that it can be passed as a document class option, so define a command that will implement both.

```

\@gls@declareoption
\newcommand*\@gls@declareoption}[2]{%
\DeclareOptionX{#1}{#2}%
\DeclareOption{#1}{#2}%
}

```

Each entry within a given glossary will have an associated number list. By default, this refers to the page numbers on which that entry has been used, but it can also refer to any counter used in the document (such as the section or equation counters). The default number list format displays the number list “as is”:

```

\glossaryentrynumbers
\newcommand*\glossaryentrynumbers}[1]{#1\gls@save@numberlist{#1}}

```

`nonumberlist` Note that the entire number list for a given entry will be passed to `\glossaryentrynumbers` so any font changes will also be applied to the delimiters. The `nonumberlist` package option suppresses the number lists (this simply redefines `\glossaryentrynumbers` to ignore its argument).

```
\gls@declareoption{nonumberlist}{%
  \renewcommand*\glossaryentrynumbers}[1]{\gls@save@numberlist{#1}}%
}
```

`savenumberlist` Provide means to store the number list for entries.

```
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{%
  \glssavenumberlistfalse}
```

`\glo@seeautonumberlist`

```
\newcommand*\glo@seeautonumberlist{}
```

`seeautonumberlist` Automatically activates number list for entries containing the see key.

```
\gls@declareoption{seeautonumberlist}{%
  \renewcommand*\glo@seeautonumberlist}{%
    \def\glo@prefix{\glsnextpages}%
  }%
}
```

`esclocations` When using `makeindex` or `xindy`, the locations may need to be adjusted to ensure they're in a format that's allowed by the indexing application. This involves a bit of hackery and isn't needed if the locations are all guaranteed to be in the correct form (or if the user is prepared to post-process the glossary file before calling the relevant indexing application, which `makeglossaries` can do) so `esclocations=false` will switch off this mechanism allowing for a faster and more stable approach. As from v4.50, the default is now false.

```
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{%
  \glseesclocationsfalse}
```

`\gls@loadlong`

```
\newcommand*\gls@loadlong{\RequirePackage{glossary-long}}
```

`nolong` This option prevents `glossary-long` from being loaded. This means that the glossary styles that use the `longtable` environment will not be available. This option is provided to reduce overhead caused by loading unrequired packages.

```
\gls@declareoption{nolong}{\renewcommand*\gls@loadlong{}}
```

`\gls@loadsuper` The `glossary-super` package isn't loaded if `supertabular` isn't installed.

```
\IfFileExists{supertabular.sty}{%
  \newcommand*\gls@loadsuper{\RequirePackage{glossary-super}}}{%
  \newcommand*\gls@loadsuper{}}
```

Don't load super styles if `xtab` has been loaded as it causes a conflict.

```
\ifpackageloaded{xtab}{\renewcommand*\gls@loadsuper{}}{}
```

`nosuper` This option prevents `glossary-super` from being loaded. This means that the glossary styles that use the `supertabular` environment will not be available. This option is provided to reduce overhead caused by loading unrequired packages.

```
\@gls@declareoption{nosuper}{\renewcommand*{\@gls@loadsuper}{}}
```

`\@gls@loadlist`

```
\newcommand*{\@gls@loadlist}{\RequirePackage{glossary-list}}
```

`nolist` This option prevents `glossary-list` from being loaded (to reduce overheads if required). Naturally, the styles defined in `glossary-list` will not be available if this option is used. If the style is still set to `list`, the default must be set to `\relax`.

```
\@gls@declareoption{nolist}{%
\renewcommand*{\@gls@loadlist}{%
\ifdefstring{\@glossary@default@style}{list}%
{\let\@glossary@default@style\relax}%
}%
}%
}
```

`\@gls@loadtree`

```
\newcommand*{\@gls@loadtree}{\RequirePackage{glossary-tree}}
```

`notree` This option prevents `glossary-tree` from being loaded (to reduce overheads if required). Naturally, the styles defined in `glossary-tree` will not be available if this option is used.

```
\@gls@declareoption{notree}{\renewcommand*{\@gls@loadtree}{}}
```

`nostyles` Provide an option to suppress all the predefined styles (in the event that the user has custom styles that are not dependent on the predefined styles).

```
\@gls@declareoption{nostyles}{%
\renewcommand*{\@gls@loadlong}{}%
\renewcommand*{\@gls@loadsuper}{}%
\renewcommand*{\@gls@loadlist}{}%
\renewcommand*{\@gls@loadtree}{}%
\let\@glossary@default@style\relax
}
```

`\glspostdescription` The description terminator is given by `\glspostdescription` (except for the 3 and 4 column styles). This is a full stop by default. The spacefactor is adjusted in case the description ends with an upper case letter. (Patch provided by Michael Pock.)

```
\newcommand*{\glspostdescription}{%
\ifglsnopostdot\else.\spacefactor\sfcode`. \fi
}
```

`nopostdot` Boolean option to suppress post description dot

```
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{%
\glsnopostdotfalse}
```

`nogroupskip` Boolean option to suppress vertical space between groups in the pre-defined styles.

```
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{}
\glsnogroupskipfalse
```

`ucmark` Boolean option to determine whether or not to use upper case in definition of `\gls glossarymark`

```
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{}

\@ifclassloaded{memoir}
{%
  \glsucmarktrue
}%
{%
  \glsucmarkfalse
}
```

`glossaryentry` If the `entrycounter` package option has been used, define a counter to number each level 0 entry. This is now defined by an internal command for consistency.

`\define@glossaryentrycounter`

```
\newcommand*{\@gls@define@glossaryentrycounter}{%
  \ifglsentrycounter
  Define the glossaryentry counter if it doesn't already exist.
  \ifundef\c@glossaryentry
  {%
    \ifx\@gls@counterwithin\@empty
    \newcounter{glossaryentry}%
    \else
    \newcounter{glossaryentry}[\@gls@counterwithin]%
    \fi
    \def\theHglossaryentry{\currentglossary.\theglossaryentry}%
  }%
  {}%
\fi
}
```

`entrycounter` Defines a counter that can be used in the standard glossary styles to number each (main) entry. If true, this will define a counter called `glossaryentry`.

```
\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\glsentrycounterfalse
```

`counterwithin` This option can be used to set a parent counter for `glossaryentry`. This option automatically sets `entrycounter=true`.

```
\define@key{glossaries.sty}{counterwithin}{%
  \renewcommand*{\@gls@counterwithin}{#1}%
  \glsentrycountertrue
  \@gls@define@glossaryentrycounter
}
```

`\@gls@counterwithin` The default value is no parent counter:

```
\newcommand*\@gls@counterwithin{}
```

`glossarysubentry` If the `subentrycounter` package option has been used, define a counter to number each level 1 entry. This is now defined by an internal command for consistency.

`\define@glossarysubentrycounter`

```
\newcommand*\@gls@define@glossarysubentrycounter}{%
```

Check if counter already defined.

```
\ifundef\c@glossarysubentry
{%
  \ifglssubentrycounter
  \ifglsubentrycounter
    \newcounter{glossarysubentry}[glossaryentry]%
  \else
    \newcounter{glossarysubentry}%
  \fi
}
```

As with `\theHglossaryentry`, this starts with `\currentglossary`. to help avoid duplicate hyper targets.

```
\def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
\fi
}%
{}%
}
```

`subentrycounter` Define a counter that can be used in the standard glossary styles to number each level 1 entry. If true, this will define a counter called `glossarysubentry`.

```
\define@boolkey{glossaries.sty}{gls}{subentrycounter}[true]{%
\glssubentrycounterfalse}
```

`\glsindexingsetting` Provide a way of finding out what the current indexing setting is. This won't pick up the `noidx` option until `\makenoidxglossaries` is encountered.

```
\newcommand*\glsindexingsetting{\ifglxsindy xindy\else makeindex\fi}
```

`\@glo@default@sorttype` Initialise default sort for `\printnoidxglossary`

```
\newcommand*\@glo@default@sorttype{standard}
```

`sort` Define the sort method: `sort=standard` (default), `sort=def` (order of definition) or `sort=use` (order of use). If no indexing required, use `sort=none` or `sort=clear`.

```
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none,clear}{%
\renewcommand*\@glo@default@sorttype}{#1}%
\csname @gls@setupsort@#1\endcsname
}
```

```
\glsprestandardsort{<sort cs>}{<type>}{<label>}
```

`\glsprestandardsort`

Allow user to hook into sort mechanism. The first argument $\langle sort\ cs \rangle$ is the temporary control sequence containing the sort value before it has been sanitized and had `makeindex/xindy` special characters escaped.

```
\newcommand*\glsprestandardsort}[3]{%
  \glsdosanitizesort
}
```

`\glo@check@sortallowed`

```
\newcommand*\glo@check@sortallowed}[1]{}
```

`\gls@setupsort@standard` Set up the macros for default sorting.

```
\newcommand*\gls@setupsort@standard}{%
```

Store entry information when it's defined.

```
\def\do@glo@storeentry{\glo@storeentry}%
```

No count register required for standard sort.

```
\def\@gls@defsortcount##1{}%
```

Sort according to sort key (`\glo@sort`) if provided otherwise sort according to the entry's name (`\glo@name`). (First argument glossary type, second argument entry label.)

```
\def\@gls@defsort##1##2{%
```

```
\ifx\@glo@sort\glsdefaultsort
```

```
\let\@glo@sort\glo@name
```

```
\fi
```

```
\let\glsdosanitizesort\gls@sanitizesort
```

```
\glsprestandardsort{\@glo@sort}{##1}{##2}%
```

```
\expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
```

```
}%
```

Don't need to do anything when the entry is used.

```
\def\@gls@setsort##1{}%
```

This sort option is allowed with `\makeglossaries` and `\makenoidxglossaries`.

```
\let\@glo@check@sortallowed@gobble
```

```
}
```

Set standard sort as the default:

```
\gls@setupsort@standard
```

`\glsortnumberfmt` Format the number used as the sort key by `sort=def` and `sort=use`. Defaults to six digit numbering.

```
\newcommand*\glsortnumberfmt[1]{%
```

```
\ifnum#1<100000 0\fi
```

```
\ifnum#1<10000 0\fi
```

```
\ifnum#1<1000 0\fi
```

```
\ifnum#1<100 0\fi
```

```
\ifnum#1<10 0\fi
```

```
\number#1%
```

```
}
```



```

\@gls@setupsort@def Set up the macros for order of definition sorting.
    \newcommand*{\@gls@setupsort@def}{%
Store entry information when it's defined.
    \def\do@glo@storeentry{\@glo@storeentry}%
Defined count register associated with the glossary.
    \def\@gls@defsortcount##1{%
    \expandafter\global
    \expandafter\newcount\csname glossary@##1@sortcount\endcsname
    }%
Increment count register associated with the glossary and use as the sort key.
    \def\@gls@defsort##1##2{%
It may be that the sort order was changed after the glossary was defined, so
check if the count register has been defined.
    \ifcsundef{glossary@##1@sortcount}%
    {\@gls@defsortcount{##1}}%
    {}%
    \expandafter\global\expandafter
    \advance\csname glossary@##1@sortcount\endcsname by 1\relax
    \expandafter\protected\def\csname glo@##2@sort\endcsname{%
    \expandafter\glssortnumberfmt
    {\csname glossary@##1@sortcount\endcsname}}%
    }%
Don't need to do anything when the entry is used.
    \def\@gls@setsort##1{%
This sort option is allowed with \makeglossaries and \makenoidxglossaries.
    \let\@glo@check@sortallowed\@gobble
    }

\@gls@setupsort@use Set up the macros for order of use sorting.
    \newcommand*{\@gls@setupsort@use}{%
Don't store entry information when it's defined.
    \let\do@glo@storeentry\@gobble
Defined count register associated with the glossary.
    \def\@gls@defsortcount##1{%
    \expandafter\global
    \expandafter\newcount\csname glossary@##1@sortcount\endcsname
    }%
Initialise the sort key to empty.
    \def\@gls@defsort##1##2{%
    \expandafter\gdef\csname glo@##2@sort\endcsname{}%
    }%
If the sort key hasn't been set, increment the counter associated with the glos-
sary and set the sort key.
    \def\@gls@setsort##1{%

```

Get the parent, if one exists

```
\protected@edef\@glo@parent{\csname glo@##1@parent\endcsname}%
```

Set the information for the parent entry if not already done.

```
\ifx\@glo@parent\@empty  
\else  
\expandafter\@gls@setsort\expandafter{\@glo@parent}%  
\fi
```

Set index information for this entry

```
\protected@edef\@glo@type{\csname glo@##1@type\endcsname}%  
\protected@edef\@gls@tmp{\csname glo@##1@sort\endcsname}%  
\ifx\@gls@tmp\@empty  
\expandafter\global\expandafter  
\advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax  
\expandafter\protected@xdef\csname glo@##1@sort\endcsname{%  
\expandafter\gls@sortnumberfmt  
\csname glossary@\@glo@type @sortcount\endcsname}}%  
\@glo@storeentry{##1}%  
\fi  
}%
```

This sort option is allowed with `\makeglossaries` and `\makenoidxglossaries`.

```
\let\@glo@check@sortallowed\@gobble  
}
```

`\@gls@setupsort@none` Slightly improves efficiency in the event that no indexing is required.

```
\newcommand*\@gls@setupsort@none{%  
\def\glsindexingsetting{none}%
```

Don't store entry index information.

```
\def\do@glo@storeentry##1{%
```

No count register required for standard sort.

```
\def\@gls@defsortcount##1{%
```

Don't modify sort value.

```
\def\@gls@defsort##1##2{%  
\expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort  
}%
```

Don't need to do anything when the entry is used.

```
\def\@gls@setsort##1{%
```

This sort option isn't allowed with `\makeglossaries` or `\makenoidxglossaries`.

```
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}  
{Option sort=none not allowed with \string##1}%  
{(Use sort=def instead)}}%  
}
```

`\@gls@setupsort@clear` Like `sort=none` but sets the sort value to empty.

```
\newcommand*\@gls@setupsort@clear{%  
\def\glsindexingsetting{none}%
```

Don't store entry index information.

```
\def\do@glo@storeentry##1{%
```

No count register required for standard sort.

```
\def\@gls@defsortcount##1{%
```

Clear sort value.

```
\def\@gls@defsort##1##2{\csgdef{glo@##2@sort}{}}%
```

Don't need to do anything when the entry is used.

```
\def\@gls@setsort##1{%
```

This sort option isn't allowed with `\makeglossaries` or `\makenoidxglossaries`.

```
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
{Option sort=clear not allowed with \string##1}%
{(Use sort=def instead)}}%
}
```

`\glsdefmain` Define the main glossary. This will be the first glossary to be displayed when using `\printglossaries`. The default extensions conflict if used with `doc`, so provide different extensions if `doc` loaded. (If these extensions are inappropriate, use `nomain` and manually define the main glossary with the desired extensions.)

```
\newcommand*\glsdefmain{%
\if@gls@docloaded
\newglossary[glg2]{main}{gls2}{glo2}{\glossaryname}%
\else
\newglossary{main}{gls}{glo}{\glossaryname}%
\fi
```

Define hook to set the toc title when translator is in use.

```
\newcommand*\gls@tr@set@main@toctitle{%
\translatelet{\glossarytoctitle}{Glossary}%
}%
}
```

Keep track of the default glossary. This is initialised to the main glossary, but can be changed if for some reason you want to make a secondary glossary the main glossary. This affects any commands that can optionally take a glossary name as an argument (or as the value of the `type` key in a key-value list). This was mainly done so that `\loadglsentries` can temporarily change `\glsdefaulttype` while it loads a file containing new glossary entries (see [subsection 2.10](#)).

`\glsdefaulttype`

```
\newcommand*\glsdefaulttype{main}
```

Keep track of which glossary the acronyms are in. This is initialised to `\glsdefaulttype`, but is changed by the acronym package option.

`\acronymtype`

```
\newcommand*\acronymtype{\glsdefaulttype}
```

`nomain` The `nomain` option suppress the creation of the main glossary.

```
\@gls@declareoption{nomain}{%  
  \let\glsdefaulttype\relax  
  \renewcommand*\glsdefmain}{}%  
}
```

`acronym` The `acronym` option sets an associated conditional which is used in [subsection 2.17](#) to determine whether or not to define a separate glossary for acronyms.

```
\define@boolkey{glossaries.sty}{gls}{acronym}[true]{%  
  \ifglsacronym  
    \renewcommand{\@gls@do@acronymsdef}{%  
      \DeclareAcronymList{acronym}%  
      \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%  
      \renewcommand*\acronymtype{acronym}%
```

Define hook to set the toc title when translator is in use.

```
\newcommand*\gls@tr@set@acronym@toctitle}{%  
  \translatelet{\glossarytoctitle}{Acronyms}%  
}%  
}%  
\else  
  \let\@gls@do@acronymsdef\relax  
\fi  
}
```

`\printacronyms` Define `\printacronyms` at the start of the document if `acronym` is set and `\printacronyms` hasn't already been defined.

```
\AtBeginDocument{%  
  \ifglsacronym  
    \providecommand*\printacronyms[1][1]{%  
      \printglossary[type=\acronymtype,#1]%  
    }  
  \fi  
}
```

`\@gls@do@acronymsdef` Set default value

```
\newcommand*\@gls@do@acronymsdef{}
```

`acronyms` Provide a synonym for `acronym=true` that can be passed via the document class options.

```
\@gls@declareoption{acronyms}{%  
  \glsacronymtrue  
  
  \def\@gls@do@acronymsdef{%  
    \DeclareAcronymList{acronym}%  
    \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%  
    \renewcommand*\acronymtype{acronym}%
```

Define hook to set the toc title when translator is in use.

```
\newcommand*\gls@tr@set@acronym@toctitle}{%  
  \translatelet{\glossarytoctitle}{Acronyms}%
```

```

    }%
  }%
}

```

`\@glsacronymlists` Comma-separated list of glossary labels indicating which glossaries contain acronyms.

```
\newcommand*{\@glsacronymlists}{}
```

`\@addtoacronymlists`

```

\newcommand*{\@addtoacronymlists}[1]{%
  \ifx\@glsacronymlists\@empty
    \protected@xdef\@glsacronymlists{#1}%
  \else
    \protected@xdef\@glsacronymlists{\@glsacronymlists,#1}%
  \fi
}

```

`\DeclareAcronymList` Identifies the named glossary as a list of acronyms and adds to the list. (Doesn't check if the glossary exists, but checks if label already in list.)

```

\newcommand*{\DeclareAcronymList}[1]{%
  \glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}

```

```
\glsIfListOfAcronyms{<label>}{<true part>}{<false part>}
```

`\glsIfListOfAcronyms`

Determines if the glossary with the given label has been identified as being a list of acronyms.

```

\newcommand{\glsIfListOfAcronyms}[1]{%
  \protected@edef\@do@gls@islistofacronyms{%
    \noexpand\@gls@islistofacronyms{#1}{\@glsacronymlists}}%
  \@do@gls@islistofacronyms
}

```

Internal command requires label and list to be expanded:

```

\newcommand{\@gls@islistofacronyms}[4]{%
  \def\gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%
    \def\@gls@before{##1}\def\@gls@after{##2}}%
  \gls@islistofacronyms,#2,#1,\@nil\end@gls@islistofacronyms
  \ifx\@gls@after\@nnil

```

Not found

```

  #4%
  \else

```

Found

```

  #3%
  \fi
}

```

```

\if@glsisacronymlist Convenient boolean.
    \newif\if@glsisacronymlist

\gls@checkisacronymlist Sets the above boolean if argument is a label representing a list of acronyms.
    \newcommand*\gls@checkisacronymlist}[1]{%
        \glsIfListOfAcronyms{#1}%
        {\@glsisacronymlisttrue}{\@glsisacronymlistfalse}%
    }

\SetAcronymLists Sets the “list of acronyms” list. Argument must be a comma-separated list of
glossary labels. (Doesn’t check at this point if the glossaries exists.)
    \newcommand*\SetAcronymLists}[1]{%
        \renewcommand*\@glsacronymlists}{#1}%
    }

acronymlists
    \define@key{glossaries.sty}{acronymlists}{%
        \DeclareAcronymList{#1}%
    }

    The default counter associated with the numbers in the glossary is stored in
    \glscounter. This is initialised to the page counter. This is used as the default
    counter when a new glossary is defined, unless a different counter is specified in
    the optional argument to \newglossary (see subsection 2.6).

\glscounter
    \newcommand{\glscounter}{page}

counter The counter option changes the default counter. (This just redefines \glscounter.)
    \define@key{glossaries.sty}{counter}{%
        \renewcommand*\glscounter}{#1}%
    }

\@gls@nohyperlist
    \newcommand*\@gls@nohyperlist{}

\GlsDeclareNoHyperList
    \newcommand*\GlsDeclareNoHyperList}[1]{%
        \ifdefempty\@gls@nohyperlist
        {%
            \renewcommand*\@gls@nohyperlist}{#1}%
        }%
        {%
            \appto\@gls@nohyperlist{,#1}%
        }%
    }

```

nohypertypes

```
\define@key{glossaries.sty}{nohypertypes}{%  
  \GlsDeclareNoHyperList{#1}%  
}
```

`\GlossariesWarning` Prints a warning message.

```
\newcommand*{\GlossariesWarning}[1]{%  
  \PackageWarning{glossaries}{#1}%  
}
```

`\GlossariesWarningNoLine` Prints a warning message without the line number.

```
\newcommand*{\GlossariesWarningNoLine}[1]{%  
  \PackageWarningNoLine{glossaries}{#1}%  
}
```

`\glosortentrieswarning` Warn user that sorting may take a long time. This is actually an informational message rather than a warning so just use `\typeout`.

```
\newcommand{\glosortentrieswarning}{%  
  \typeout{Using TeX to sort glossary entries---this may  
  take a while}%  
}
```

`nowarn` Define package option to suppress warnings

```
\@gls@declareoption{nowarn}{%  
  \if@gls@debug  
    \GlossariesWarning{Warnings can't be suppressed in debug mode}%  
  \else  
    \renewcommand*{\GlossariesWarning}[1]{}%  
    \renewcommand*{\GlossariesWarningNoLine}[1]{}%  
    \renewcommand*{\glosortentrieswarning}{}%  
    \renewcommand*{\@gls@missinglang@warn}[2]{}%  
  \fi  
}
```

`\@gls@missinglang@warn` Missing language warning.

```
\newcommand*{\@gls@missinglang@warn}[2]{%  
  \PackageWarningNoLine{glossaries}%  
  {No language module detected for '#1'.\MessageBreak  
  Language modules need to be installed separately.\MessageBreak  
  Please check on CTAN for a bundle called\MessageBreak  
  'glossaries-#2' or similar}%  
}
```

`nolangwarn` Suppress warning if language support not found.

```
\@gls@declareoption{nolangwarn}{%  
  \renewcommand*{\@gls@missinglang@warn}[2]{}%  
}
```

`\@gls@warnonglossdefined` Issue a warning if overriding `\printglossary`

```

\newcommand*\@gls@warnonglossdefined{%
  \GlossariesWarning{Overriding \string\printglossary}%
}

```

`\@gls@warnontheGLOSSdefined` Issue a warning if overriding `theglossary`

```

\newcommand*\@gls@warnontheGLOSSdefined{%
  \GlossariesWarning{Overriding 'theglossary' environment}%
}

```

`noredefwarn` Suppress warning on redefinition of `\printglossary`

```

\@gls@declareoption{noredefwarn}{%
  \renewcommand*\@gls@warnonglossdefined{}%
  \renewcommand*\@gls@warnontheGLOSSdefined{}%
}

```

As from version 3.08a, the only information written to the external glossary files are the label and sort values. Therefore, now, the only `sanitize` option that makes sense is the one for the sort key. so the `sanitize` option is now deprecated and there is only a `sanitizesort` option.

`\@gls@sanitizedesc` `\@gls@sanitizedesc` was deprecated in v3.08a and removed in v4.50.

`\glssetexpandfield`

```
\glssetexpandfield{<field>}
```

Sets field to always expand.

```

\newcommand*\glssetexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@expand@field{##1}{#1}{##2}%
  }%
}

```

`\glssetnoexpandfield`

```
\glssetnoexpandfield{<field>}
```

Sets field to never expand.

```

\newcommand*\glssetnoexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@noexpand@field{##1}{#1}{##2}%
  }%
}

```

`\gls@assign@type@field` The type must always be expandable.

```
\glssetexpandfield{type}
```

`\gls@assign@desc@field` The description is not expanded by default:

```
\glssetnoexpandfield{desc}
```



```

\gls@assign@desclplural@field
    \glssetnoexpandfield{desclplural}

\@gls@sanitizename
    \newcommand*{\@gls@sanitizename}{}

\gls@assign@name@field Don't expand name by default.
    \glssetnoexpandfield{name}

\@gls@sanitizesymbol
    \newcommand*{\@gls@sanitizesymbol}{}

\@gls@assign@symbol@field Don't expand symbol by default.
    \glssetnoexpandfield{symbol}

\@gls@assign@symbolplural@field
    \glssetnoexpandfield{symbolplural}

    Sanitizing stuff:

\@gls@sanitizesort
    \newcommand*{\@gls@sanitizesort}{%
        \ifglssanitizesort
            \@gls@sanitizesort
        \else
            \@gls@nosanitizesort
        \fi
    }

\@@gls@sanitizesort
    \newcommand*\@@gls@sanitizesort{%
        \@onelevel@sanitize\@glo@sort
    }

\@@gls@nosanitizesort
    \newcommand*\@@gls@nosanitizesort{}

\@@gls@noidx@sanitizesort Remove braces around first character (if present) before sanitizing.
    \newcommand*\@gls@noidx@sanitizesort{%
        \ifdefvoid\@glo@sort
            {}%
            {%
                \expandafter\@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
            }%
        }
    \def\@@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
        \def\@glo@sort{#1#2}%
        \@onelevel@sanitize\@glo@sort
    }

```

`\@@gls@noidx@nosanitizesort`

```
\newcommand*{\@@gls@noidx@nosanitizesort}{%
  \ifdefvoid\@glo@sort
  }%
  {%
    \expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
  \bgroup
  \glsnoidxstripaccents
  \protected@xdef\@glo@sort{#1#2}%
  \egroup
  \let\@glo@sort\@@glo@sort
}
```

`\glsnoidxstripaccents` This strips accents by redefining the standard accent commands to just do their argument. (This will be localised since `\glsnoidxstripaccents` is used within a group.) Anything outside this standard set really shouldn't be using `\makenoidxglossaries`. It's much better to use `xindy` or `bib2gls` with the correct language setting.

```
\newcommand*\glsnoidxstripaccents{%
  \let\IeC\@firstofone
  \let\add@accent@\@secondoftwo
  \let\@text@composite@x\@secondoftwo
  \let\@tabacckludge\@secondoftwo
  \expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
  \let'\@firstofone
  \let'\@firstofone
  \let^\@firstofone
  \let"@@firstofone
  \let\u\@firstofone
  \let\t\@firstofone
  \let\d\@firstofone
  \let\r\@firstofone
  \let=\@firstofone
  \let.\@firstofone
  \let~\@firstofone
  \let\v\@firstofone
  \let\H\@firstofone
  \let\c\@firstofone
  \let\b\@firstofone

  \let\a\@secondoftwo
  \def\AE{AE}%
  \def\ae{ae}%
}
```

```

\def\OE{OE}%
\def\oe{oe}%
\def\AA{AA}%
\def\aa{aa}%
\def\L{L}%
\def\l{l}%
\def\O{O}%
\def\o{o}%
\def\SS{SS}%
\def\ss{ss}%
\def\th{th}%

\def\TH{TH}%
\def\dh{dh}%
\def\DH{DH}%
}

```

Need to check if the LaTeX kernel is at least version 2019/10/01 as that changes the way that UTF-8 characters expand.

```

\ifl@t@r\fmtversion{2019/10/01}
{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}

```

`\@gls@removedoption`

```

\newcommand{\@gls@removedoption}[1]{%
\PackageError{glossaries}{obsolete package option ‘#1’ has been
removed. Rollback required or use a newer alternative}
{Try \string\usepackage[#1]{glossaries}[=v4.46]}%
}

```

Before defining the sanitize package option, The key-value list for the sanitize value needs to be defined. These are all boolean keys. If they are not given a value, assume true.

```

\define@boolkey[gls]{sanitize}{description}[true]{%
\GlossariesWarning{sanitize={description} package option deprecated}%
\ifgls@sanitize@description
\glssetnoexpandfield{desc}%
\glssetnoexpandfield{descplural}%
\else
\glssetexpandfield{desc}%
\glssetexpandfield{descplural}%
\fi
}

\define@boolkey[gls]{sanitize}{name}[true]{%
\GlossariesWarning{sanitize={name} package option deprecated}%
\ifgls@sanitize@name
\glssetnoexpandfield{name}%

```

```

\else
  \glsssetexpandfield{name}%
\fi
}
\define@boolkey[glS]{sanitize}{symbol}[true]{%
  \GlossariesWarning{sanitize={symbol} package option deprecated}%
  \ifglS@sanitize@symbol
    \glsssetnoexpandfield{symbol}%
    \glsssetnoexpandfield{symbolplural}%
  \else
    \glsssetexpandfield{symbol}%
    \glsssetexpandfield{symbolplural}%
  \fi
}

```

sanitizesort

```

\define@boolkey{glossaries.sty}[glS]{sanitizesort}[true]{%
  \ifglSSsanitizesort
    \glsssetnoexpandfield{sortvalue}%
    \renewcommand*{\@glS@noidx@setsanitizesort}{%
      \glsssanitizesorttrue
      \glsssetnoexpandfield{sortvalue}%
    }%
  \else
    \glsssetexpandfield{sortvalue}%
    \renewcommand*{\@glS@noidx@setsanitizesort}{%
      \glsssanitizesortfalse
      \glsssetexpandfield{sortvalue}%
    }%
  \fi
}

```

Default setting:

```

\glsssanitizesorttrue
\glsssetnoexpandfield{sortvalue}%

```

`\@glS@noidx@setsanitizesort` Default behaviour for `\makenoidxglossaries` is `sanitizesort=false`.

```

\newcommand*{\@glS@noidx@setsanitizesort}{%
  \glsssanitizesortfalse
  \glsssetexpandfield{sortvalue}%
}

```

```

\define@choicekey[glS]{sanitize}{sort}{true,false}[true]{%
  \setbool{glSSsanitizesort}{#1}%
  \ifglSSsanitizesort
    \glsssetnoexpandfield{sortvalue}%
  \else
    \glsssetexpandfield{sortvalue}%
  \fi
  \GlossariesWarning{sanitize={sort} package option

```

```

    deprecated. Use sanitizesort instead}%
}

```

`sanitize`

```

\define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
  \ifthenelse{\equal{#1}{none}}{%
    {%
      \GlossariesWarning{sanitize package option deprecated}%
      \glssetexpandfield{name}%
      \glssetexpandfield{symbol}%
      \glssetexpandfield{symbolplural}%
      \glssetexpandfield{desc}%
      \glssetexpandfield{descplural}%
    }%
  }%
  \setkeys[gls]{sanitize}{#1}%
}%
}

```

`\ifglstranslate` As from version 3.13a, the translator package option is a choice rather than boolean option so now need to define conditional:

```

\newif\ifglstranslate

```

`\@gls@notranslatorhook` `\@gls@notranslatorhook` has been removed.

`\@gls@usetranslator`

```

\newcommand*\@gls@usetranslator{%
polyglossia tricks \@ifpackageloaded into thinking that babel has been loaded,
so check for polyglossia as well.
  \@ifpackageloaded{polyglossia}%
  {%
    \let\glsifusetranslator\@secondoftwo
  }%
  {%
    \@ifpackageloaded{babel}%
    {%
      \IfFileExists{translator.sty}%
      {%
        \RequirePackage{translator}%
        \let\glsifusetranslator\@firstoftwo
      }%
    }%
  }%
  {}%
}%
}

```

`\glsifusedtranslatordict` Checks if given translator dictionary has been loaded.

```

\newcommand{\glsifusedtranslatordict}[3]{%
  \glsifusetranslator
  {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
  {#3}%
}

```

`nottranslate` Provide a synonym for `translate=false` that can be passed via the document class.

```

\@gls@declareoption{nottranslate}{%
  \glstranslatefalse
  \let\@gls@usetranslator\relax
  \let\glsifusetranslator\@secondoftwo
}

```

`translate` Define `translate` option. If false don't set up multi-lingual support.

```

\define@choicekey{glossaries.sty}{translate}{%
  [\gls@translate@val\gls@translate@nr]%
  {true,false,babel}[true]%
  {%
    \ifcase\gls@translate@nr\relax
      \glstranslatetrue
      \renewcommand*\@gls@usetranslator{%
        \@ifpackageloaded{polyglossia}%
        {%
          \let\glsifusetranslator\@secondoftwo
        }%
        {%
          \@ifpackageloaded{babel}%
          {%
            \IfFileExists{translator.sty}%
            {%
              \RequirePackage{translator}%
              \let\glsifusetranslator\@firstoftwo
            }%
            {}%
          }%
          {}%
        }%
      }%
    \or
      \glstranslatefalse
      \let\@gls@usetranslator\relax
      \let\glsifusetranslator\@secondoftwo
    \or
      \glstranslatetrue
      \let\@gls@usetranslator\relax
      \let\glsifusetranslator\@secondoftwo
    \fi
  }
}

```

Set the default value:

```
\glstranslatefalse
\let\glsifusetranslator\@secondoftwo
\@ifpackageloaded{translator}%
{%
  \glstranslatetrue
  \let\glsifusetranslator\@firstoftwo
}%
{%
  \for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
  {
    \@ifpackageloaded{\gls@thissty}%
    {%
      \glstranslatetrue
      \@endfortrue
    }%
    {}%
  }
}
```

`\@gls@requested@languages`

```
\newcommand{\@gls@requested@languages}{}
```

languages Identify required languages independently of any language packages and switches off automatic loading of translator.

```
\define@key{glossaries.sty}{languages}{%
  \renewcommand{\@gls@requested@languages}{#1}%
  \glstranslatetrue
  \let\@gls@usetranslator\relax
  \let\glsifusetranslator\@secondoftwo
}
```

locales Provide synonym.

```
\define@key{glossaries.sty}{locales}{%
  \renewcommand{\@gls@requested@languages}{#1}%
  \glstranslatetrue
  \let\@gls@usetranslator\relax
  \let\glsifusetranslator\@secondoftwo
}
```

indexonlyfirst Set whether to only index on first use.

```
\define@boolkey{glossaries.sty}[gls]{indexonlyfirst}[true]{}
\glsindexonlyfirstfalse
```

hyperfirst Set whether or not terms should have a hyperlink on first use.

```
\define@boolkey{glossaries.sty}[gls]{hyperfirst}[true]{}
\glshyperfirsttrue
```

`\@gls@setacrstyle` Keep track of whether an acronym style has been set (for the benefit of `\setupglossaries`):

```

\newcommand*{\@gls@setacrstyle}{%

```

`footnote` Set the long form of the acronym in footnote on first use.

```

\define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{%
\ifglsacrfootnote
\@gls@removedoption{footnote}%
\fi
}

```

`description` Allow acronyms to have a description (needs to be set using the `description` key in the optional argument of `\newacronym`).

```

\define@boolkey{glossaries.sty}[glsacr]{description}[true]{%
\ifglsacrdescription
\@gls@removedoption{description}%
\fi
}

```

`smallcaps` Define `\newacronym` to set the short form in small capitals.

```

\define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{%
\ifglsacrsmallcaps
\@gls@removedoption{smallcaps}%
\fi
}

```

`smaller` Define `\newacronym` to set the short form using `\smaller` which obviously needs to be defined by loading the appropriate package.

```

\define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{%
\ifglsacrsmaller
\@gls@removedoption{smaller}%
\fi
}

```

`dua` Define `\newacronym` to always use the long forms (i.e. don't use acronyms)

```

\define@boolkey{glossaries.sty}[glsacr]{dua}[true]{%
\ifglsacrdua
\@gls@removedoption{dua}%
\fi
}

```

`shortcuts` Define acronym shortcuts.

```

\define@boolkey{glossaries.sty}[glsacr]{shortcuts}[true]{%

```

`\glsorder` Stores the glossary ordering. This may either be “word” or “letter”. This passes the relevant information to `makeglossaries`. The default is word ordering.

```

\newcommand*{\glsorder}{word}

```


`\@glsorder` The ordering information is written to the auxiliary file for `makeglossaries`, so ignore the auxiliary information.

```
\newcommand*\@glsorder}[1]{}
```

`order`

```
\define@choicekey{glossaries.sty}{order}{word,letter}{%
\def\glsorder{#1}}
```

`\ifglsxindy` Provide boolean to determine whether `xindy` or `makeindex` will be used to sort the glossaries.

```
\newif\ifglsxindy
```

The default is `makeindex`:

```
\glsxindyfalse
```

`makeindex` Define package option to specify that `makeindex` will be used to sort the glossaries:

```
\@gls@declareoption{makeindex}{\glsxindyfalse}
```

The `xindy` package option may have a value which in turn can be a key=value list. First define the keys for this sub-list. The boolean `glsnumbers` determines whether to automatically add the `glsnumbers` letter group.

```
\define@boolkey[gls]{xindy}{glsnumbers}[true]{%
\gls@xindy@glsnumberstrue
```

`\@xdy@main@language` Define what language to use for each glossary type (if a language is not defined for a particular glossary type the language specified for the main glossary is used.)

```
\def\@xdy@main@language{\languagename}%
```

`\gls@xdy@set@lang` If the `nomain` option has been used, need to check `\glsdefaulttype` before falling back on main. This will set `\@gls@langmod` for the given glossary.

```
\newcommand*\gls@xdy@set@lang}[1]{%
\ifcsundef{@xdy@#1@language}%
{%
\ifcsundef{@xdy@\glsdefaulttype @language}
{\let\@gls@langmod\@xdy@main@language}%
{\letcs\@gls@langmod{@xdy@\glsdefaulttype @language}}%
}%
{\letcs\@gls@langmod{@xdy@#1@language}}%
}
```

Define key to set the language

```
\define@key[gls]{xindy}{language}{\def\@xdy@main@language{#1}}
```

`\gls@codepage` Define the code page. If `\inputencodingname` is defined use that, otherwise initialise to utf8.

```
\ifdefvoid\inputencodingname
{\def\gls@codepage{utf8}}%
{\def\gls@codepage{\inputencodingname}}
```

Define a key to set the code page.

```
\define@key[gls]{xindy}{codepage}{\def\gls@codepage{#1}}
```

`xindy` Define package option to specify that `xindy` will be used to sort the glossaries:

```
\define@key[glossaries.sty]{xindy}[]{%
\glsxindytrue
\setkeys[gls]{xindy}{#1}%
}
```

`xindygloss` Provide a synonym for `xindy` that can be passed via the document class options.

```
\@gls@declareoption{xindygloss}{%
\glsxindytrue
}
```

`xindynoglsnumbers` Provide a synonym for `xindy=glsnumbers=false` that can be passed via the document class options.

```
\@gls@declareoption{xindynoglsnumbers}{%
\glsxindytrue
\gls@xindy@glsnumbersfalse
}
```

`\@domakeglossaries`

```
\providecommand{\@domakeglossaries}[1]{\if@filesw #1\fi}
```

`disablemakegloss` Provide a way of disabling `\makeglossaries`. For example, if a class or package explicitly uses `\makeglossaries`. This is a valueless option to allow it to be passed through the document class option list.

```
\@gls@declareoption{disablemakegloss}{%
\ifdefequal\makeglossaries\@no@makeglossaries
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect
(\string\makenoidxglossaries\space has already been used)}%
}%
{%
\ifdefequal\@makeglossary\@gobble
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect
(\string\makeglossaries\space has already been used)}%
}%
}%
\renewcommand{\@domakeglossaries}[1]{%
\PackageInfo{glossaries}{\string\makeglossaries\space and
\string\makenoidxglossaries\space have been disabled}%
}
```

```

    }%
  }%
}

```

restoremakeloss Cancel the effect of disablemakegloss.

```

\@gls@declareoption{restoremakeloss}{%
  \ifdefequal\makeglossaries\@no@makeglossaries
  {%
    \GlossariesWarning{Option ‘restoremakeloss’ has no effect
      (\string\makenoidxglossaries\space has already been used)}%
  }%
  {%
    \ifdefequal\@makeglossary\@gobble
    {%
      \GlossariesWarning{Option ‘restoremakeloss’ has no effect
        (\string\makeglossaries\space has already been used)}%
    }%
    {%
      \PackageInfo{glossaries}{\string\makeglossaries\space and
        \string\makenoidxglossaries\space have been restored}%
      \let\@domakeglossaries\@firstofone
    }%
  }%
}

```

\@do@write@glslabels

```

\newcommand*{\@do@write@glslabels}{%
  \AtEndDocument{\@do@write@glslabels}%
  \let\@do@write@glslabels\relax
}

```

\@@do@write@glslabels

```

\newcommand*{\@@do@write@glslabels}{%
  \newwrite\@gls@labelsfile
  \immediate\openout\@gls@labelsfile=\jobname.glslabels
  \protected@edef\@gls@fulllist{\@glo@types}%
  \ifdefempty{\@ignored@glossaries}{}%
  {\protected@eappto\@gls@fulllist{,\@ignored@glossaries}}%
  \forallglsentries[\@gls@fulllist]{\@glsentry}%
  {%
    \ifdefempty{\@glsentry}{}%
    {%
      \ifx\do@write@glslabelsname\@gobble
        \def\@gls@thisentryname{%
      }%
      \else
        \letcs{\@gls@thisentryname}{glo@\@glsdetoklabel{\@glsentry}@name}%
        \@onelevel@sanitize\@gls@thisentryname
      \fi
    }%
  }%
}

```

```

        \immediate\write\@gls@labelsfile
        {\@glsentry\do@write@glslabelsname{\@gls@thisentryname}}%
    }%
}
\immediate\closeout\@gls@labelsfile
}

```

`\do@write@glslabelsname` Support for additionally writing the name as well as the label. Let to `\@gobble` if name not required and to `\@write@glslabelsname` if required.

```
\newcommand*{\do@write@glslabelsname}[1]{}
```

`\@write@glslabelsname`

```

\begingroup
\catcode'\^^I12
\gdef\@write@glslabelsname#1{\^^I#1}
\endgroup

```

`writeglslabels` This option will write all entry labels (including those in ignored glossaries) to the file `\jobname.glslabels`. This file may be used by text editors for label auto-completion.

```

\@gls@declareoption{writeglslabels}{\do@write@glslabels
\let\do@write@glslabelsname\@gobble}

```

`writeglslabelnames` As the above but also writes the name.

```

\@gls@declareoption{writeglslabelnames}{\do@write@glslabels
\let\do@write@glslabelsname\@write@glslabelsname}

```

`\ifglsautomake`

```
\newif\ifglsautomake
```

`\gls@automake@nr`

```
\newcommand{\gls@automake@nr}{1}
```

`\gls@automake@makegloss` Provide a way to allow `makeglossaries` or `makeglossaries-lite` run via `automake` instead of explicit calls to `makeindex` or `xindy`. Extended by `glossaries-extra` to also call `bib2gls` if applicable.

```
\newcommand{\gls@automake@makegloss}{}

```

`\gls@automake@option`

```
\newcommand{\gls@automake@option}{automake=\gls@automake@val}
```

`\@gls@do@automake@err` NB `glossaries-extra v1.14` defines similar command `\@gls@doautomake@err`.

```

\newcommand*{\@gls@do@automake@err}{%
\PackageError{glossaries}{You must use
\string\makeglossaries\space with \gls@automake@option}
{%
    Either remove the \gls@automake@option\space setting or
    add \string\makeglossaries\space to your document preamble.%
}%
}%

```

`automake` If this setting is on, automatically run `makeindex/xindy` at the end of the document. Must be used with `\makeglossaries` (unless `glossaries-extra` is loaded with `record` option). Default is false. As from v4.42, this is now a choice rather than boolean key.

```
\define@choicekey{glossaries.sty}{automake}%
  [\gls@automake@val\gls@automake@nr]{true,false,immediate,makegloss,lite,delayed}[immediate]{%
  \ifcase\gls@automake@nr\relax
    \GlossariesWarning{automake=true deprecated. Use automake=delayed for a delayed write}%
```

case 0: true (same as delayed)

```
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\or
```

case 1: false

```
\glsautomakefalse
\renewcommand{\gls@automake@makegloss}{}%
\or
```

case 2: immediate

```
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\or
```

case 3: makegloss

```
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{makeglossaries \@gls@run@output@dir{-d}}%
\or
```

case 4: lite

```
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
\or
```

case 5: delayed

```
\def\gls@automake@nr{0}%
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\fi
\renewcommand{\gls@automake@option}{automake=\gls@automake@val}%
\ifglsautomake
  \let\@gls@doautomake\@gls@do@automake@err
\else
  \renewcommand*{\@gls@doautomake}{}%
\fi
```

```
}
\glsautomakefalse
```

`\@gls@doautomake`

```
\newcommand*{\@gls@doautomake}{%
\AtEndDocument{\@gls@doautomake}
```

`automakegloss` Provide a synonym for `automake=makegloss` that can be passed via the document class options.

```
\@gls@declareoption{automakegloss}{%
  \glsautomaketrue
  \renewcommand{\gls@automake@option}{automakegloss}%
  \def\gls@automake@val{makegloss}%
  \def\gls@automake@nr{3}%
  \renewcommand{\gls@automake@makegloss}{makeglossaries \@gls@run@output@dir{-d}}%
  \let\@gls@doautomake\@gls@do@automake@err
}
```

`automakeglosslite` Provide a synonym for `automake=lite` that can be passed via the document class options.

```
\@gls@declareoption{automakeglosslite}{%
  \glsautomaketrue
  \renewcommand{\gls@automake@option}{automakeglosslite}%
  \def\gls@automake@val{lite}%
  \def\gls@automake@nr{4}%
  \renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
  \let\@gls@doautomake\@gls@do@automake@err
}
```

`savewrites` The `savewrites` package option is provided to save on the number of write registers.

```
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
  \ifgls@savewrites
    \renewcommand*{\gls@writefiles}{\@gls@writefiles}%
  \else
    \let\gls@writefiles\@empty
  \fi
}
```

Set default:

```
\gls@savewritesfalse
\let\gls@writefiles\@empty
```

`compatible-3.07` Rollback should now be used for backward-compatibility.

```
\define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{%
  \ifbool{gls@compatible-3.07}%
  {%
    \@gls@removedoption{compatible-3.07}%
  }%
  {}%
}
\boolfalse{gls@compatible-3.07}
```

`compatible-2.07` Rollback should now be used for backward-compatibility.

```
\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
```

```

\ifbool{glscompatible-2.07}%
{%
  \@gls@removedoption{compatible-2.07}%
}%
{}%
}
\boolfalse{glscompatible-2.07}

```

`\gls@original@makeglossary` Store the original definition.

```
\let\gls@original@makeglossary\makeglossary
```

`\gls@original@glossary` Store the original definition.

```
\let\gls@original@glossary\glossary
```

`\makeglossary` The `\makeglossary` command is redefined to be identical to `\makeglossaries`. (This is done partly to reinforce the message that you must either use `\@makeglossary` for all the glossaries or for none of them, but is also a legacy from the old glossary package.) As from v4.50 this command will only be redefined if `kernelglossredefs` is used.

`\@gls@override@glossary`

```

\newcommand*\@gls@override@glossary}[1][main]{%
  \GlossariesWarning{Use of \string\glossary\space with
glossaries.sty is deprecated. \MessageBreak Indexing should be performed
with the user level \MessageBreak commands, such as \string\gls\space or
\string\glsadd. If you need the \MessageBreak original definition of
\string\glossary\space use the package \MessageBreak options
kernelglossredefs=false (to restore the \MessageBreak former definition of
\string\glossary) and nomain (if the \MessageBreak file extensions cause a
conflict)}}%
  \gls@glossary{#1}%
}

```

In v4.10, the redefinition of `\glossary` was removed since it was never intended as a user level command (and wasn't documented in the user manual), however it seems there are packages that have hacked the internal macros used by `glossaries` and no longer work with this redefinition removed, so it's been restored in v4.11 but is not used at all by `glossaries`. (This may be removed or moved to a compatibility mode in future.) As from v4.41, the use of `\glossary` now triggers a warning. The package option `kernelglossredefs=nowarn` may be used to remove the warning, but it's better not to use `\glossary`.

`\glossary` As from v4.50 this command will only be redefined if `kernelglossredefs` is used.

`kernelglossredefs` The `glossaries` package redefines the kernel commands `\makeglossary` and `\glossary` as a legacy action from the former `glossary` package. In hindsight that wasn't a good idea as it's possible that the `glossaries` package may need to be used with another class or package that needs these commands. Neither of these commands are documented in the main user manual and their use is

not encouraged. The preferred commands are `\makeglossaries` (to open all associated glossary files) and `\gls`, `\glstext` etc or `\glsadd` for indexing.

As from v4.50, the default is `kernelglossredefs=false`.

```
\define@choicekey{glossaries.sty}{kernelglossredefs}%
[\gls@kernelglossredefs@val\gls@kernelglossredefs@nr]{true,false,nowarn}[true]%
{%
  \ifcase\gls@kernelglossredefs@nr\relax
  \def\glossary{\@gls@override@glossary}%
  \def\makeglossary{%
    \GlossariesWarning{Use of \string\makeglossary\space with
      glossaries.sty is deprecated. Use \string\makeglossaries\space
      instead. If you need the original definition of
      \string\makeglossary\space use the package options
      kernelglossredefs=false (to prevent redefinition of
      \string\makeglossary) and nomain (if the file extensions cause a
      conflict)}%
    \makeglossaries
  }%
  \or
  \let\glossary\gls@original@glossary
  \let\makeglossary\gls@original@makeglossary
  \or
  \def\makeglossary{\makeglossaries}%
  \renewcommand*{\@gls@override@glossary}[1][main]{%
    \gls@glossary{##1}%
  }%
  \fi
}
```

`symbols` Create a “symbols” glossary type

```
\@gls@declareoption{symbols}{%
  \let\@gls@do@symbolsdef\@gls@symbolsdef
}
```

Default is not to define the symbols glossary:

```
\newcommand*{\@gls@do@symbolsdef}{}%
```

`\@gls@symbolsdef`

```
\newcommand*{\@gls@symbolsdef}{%
  \newglossary[slg]{symbols}{sls}{slo}{\glsymbolsgroupname}%
  \newcommand*{\printsymbols}[1][ ]{\printglossary[type=symbols,##1]}%
```

Define hook to set the toc title when translator is in use.

```
\newcommand*{\gls@tr@set@symbols@toctitle}{%
  \translatelet{\glossarytoctitle}{Symbols (glossaries)}%
}%
}%
```

`numbers` Create a “symbols” glossary type

```
\@gls@declareoption{numbers}{%
```



```

\let\@gls@do@numbersdef\@gls@numbersdef
}

```

Default is not to define the numbers glossary:

```

\newcommand*\@gls@do@numbersdef\{}

```

`\@gls@numbersdef`

```

\newcommand*\@gls@numbersdef\{%
\newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
\newcommand*\@printnumbers}[1][\]{\printglossary[type=numbers,##1]}%

```

Define hook to set the toc title when translator is in use.

```

\newcommand*\@gls@tr@set@numbers@toctitle\{%
\translatelet{\glossarytoctitle}{Numbers (glossaries)}%
}%
}%

```

`index` Create an “index” glossary type

```

\@gls@declareoption{index}\{%
\ifx\@gls@do@indexdef\@empty
\let\@gls@do@indexdef\@gls@indexdef
\fi
}

```

`noglossaryindex` Counteract index if it happens to be globally used in the document class.

```

\@gls@declareoption{noglossaryindex}\{%
\let\@gls@do@indexdef\relax
}

```

Default is not to define index glossary:

```

\newcommand*\@gls@do@indexdef\{}

```

`\@gls@indexdef` `\indexname` isn’t set by glossaries.

```

\newcommand*\@gls@indexdef\{%
\newglossary[ilg]{index}{ind}{idx}{\indexname}%
\newcommand*\@printindex}[1][\]{\printglossary[type=index,##1]}%
\newcommand*\@newterm}[2][\]{%
\newglossaryentry{##2}%
{type={index},name={##2},description={\nopostdesc},##1}}
\let\@gls@do@indexdef\relax
}%

```

`mfirstuc` Similar to `mfirstuc`’s expanded and unexpanded package options. Not providing the other option.

```

\define@choicekey{glossaries.sty}{mfirstuc}%
[\gls@mfirstuc@val\gls@mfirstuc@nr]%
{expanded,unexpanded}%
{%
\ifcase\gls@mfirstuc@nr
% expanded

```

```

\renewcommand*\glsmakefirstuc}[1]{\MFUsentencecase{#1}}%
\or
% unexpanded
\renewcommand*\glsmakefirstuc}[1]{\MFUsentencecase{\unexpanded{#1}}}%
\fi
}

```

Process package options. First process any options that have been passed via the document class.

```

\@for\CurrentOption :=\@declaredoptions\do{%
\ifx\CurrentOption\@empty
\else
\@expandtwoargs
\in@ {,\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
\ifin@
\@use@option
\expandafter \let\csname ds@\CurrentOption\endcsname\@empty
\fi
\fi
}

```

Now process options passed to the package:

```
\ProcessOptionsX
```

No longer load backward compatibility stuff as from v4.50.

If LuaLaTeX and automake set, need to load shellesc. Note that if automake is passed to `\setupglossaries` instead, then shellesc will need to be loaded explicitly. Unless the L^AT_EX kernel is new enough to support l3sys shell commands

```
\@gls@runshell
```

```

\newcommand*\@gls@runshell}[1]{%
\PackageWarning{glossaries}{Ignoring attempt to run command ‘#1’}%
}

```

```
\@gls@run@unrestricted@shell
```

```
\newcommand*\@gls@run@unrestricted@shell}[1]{\@gls@runshell{#1}}
```

```
\@gls@run@output@dir
```

```
\@gls@run@output@dir{<switch>}
```

This should expand to nothing if no output directory or to `<switch> <dir>` otherwise, where `<switch>` is the application’s switch that specifies the output directory and `<dir>` is the output directory.

```
\newcommand*\@gls@run@output@dir}[1]{}
```

For the case where there’s no switch for the directory and the file name needs to include the output path:

`\@gls@run@output@file`

```
\@gls@run@output@file{\filename}
```

This should expand to `\filename` or `\dir\sep\filename` where `\dir` is the output directory and `\sep` is the directory separator. This is provided for direct calls to `makeindex` or `xindy`.

```
\newcommand*\@gls@run@output@file}[1]{#1}
```

```
\ifglsautomake
```

Only load `shellesc` if LuaLaTeX and delayed write or old kernel.

```
\newcommand*\@gls@loadshellesc{\ifdef\directlua{\RequirePackage{shellesc}}{}}%
\ExplSyntaxOn
\sys_if_shell:TF
{
  \sys_if_shell_unrestricted:F
  {
    \cs_set:Npn \@gls@run@unrestricted@shell #1
    {
```

Warn rather than error in case using restricted mode for draft document.

```
\GlossariesWarningNoLine
{
  Unrestricted ~ shell ~ required ~ for ~ current ~
  automake ~ setting. ~
  Either ~ remove ~ automake ~ option ~ and ~ run ~ the ~
  system ~ command ~ outside ~ of ~ TeX ~ or ~ enable ~
  unrestricted ~ mode ~ (but ~ be ~ aware ~ of ~ the ~
  security ~ risks ~ in ~ doing ~ so)
}
```

Try running anyway in case some debugging tool will parse the log file afterwards.

```
\@gls@runshell { #1 }
}
}
```

Try to determine if the output directory has been set. This requires at least TeX Live 2024 and is only applicable for `makeglossaries` and `bib2gls`. Unsupported with `makeglossaries-lite` and for direct calls to `makeindex` and `xindy`. In the case of `makeindex`, the directory can be included in the reference for the input, output and style files, but the OS directory divider needs to be determined. This doesn't seem to work for `xindy` as it can't find the style module.

TODO: allow for the possibility that the environment variable has already been found? For now, just test if the token list variable has already been defined. Possible problems with user setting the variable: awkward characters need to be dealt with for one document but expansion may be required for another. For example, if the directory divider is a backslash and needs to be included in the path or if some other active character is included in the path name then the token list variable will need to have its contents converted. On the other hand,

the user may have their own placeholder command in which case the token list variable provided here needs to expand.

```

\tl_if_exist:NF \l_glossaries_output_dir_tl
{
  \tl_new:N \l_glossaries_output_dir_tl
  \cs_if_exist:NT \sys_get_shell:nnN
  {
    \sys_get_shell:nnNF
    { kpsewhich ~ --var-value=TEXMF_OUTPUT_DIRECTORY }
    { }
    \l_glossaries_output_dir_tl
    {
      \tl_clear:N \l_glossaries_output_dir_tl
    }
  }
}
\tl_if_empty:NF \l_glossaries_output_dir_tl
{
  \cs_set:Npn \@gls@run@output@dir #1
  {
    #1 ~ \glsquote { \l_glossaries_output_dir_tl }
  }
}

```

Check platform to determine directory divider. NB \jobname may include quotes so don't include argument inside \glsquote. Tested on Linux only.

```

\sys_if_platform_windows:TF
{
  \cs_set:Npn \@gls@run@output@file #1
  {
    \glsquote { \l_glossaries_output_dir_tl } \glsbackslash #1
  }
}
{
  \cs_set:Npn \@gls@run@output@file #1
  {
    \glsquote { \l_glossaries_output_dir_tl } / #1
  }
}
\int_if_zero:nTF { \gls@automake@nr }
{

```

Delayed write

```

\cs_set:Npn \@gls@runshell #1 { \exp_args:Ne \write18 { #1 } }
}
{

```

Immediate write

```

\cs_set:Npn \@gls@runshell #1 { \sys_shell_now:e { #1 } }
\cs_set_eq:NN \@gls@loadshellesc \relax
}

```

```

}
{
  \cs_set_eq:NN \@gls@loadshellesc \relax
  \legacy_if_set_false:n { glsautomake }
  \PackageWarning { glossaries }
  {
    Can't ~ implement ~ automake: ~ no ~ shell
  }
}
\ExplSyntaxOff
\@gls@loadshellesc
\fi

```

`\setupglossaries` Provide way to set options after package has been loaded. However, some options must be set before `\ProcessOptionsX`, so they have to be disabled:

```

\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,notranslate,languages,locales,nolong,nosuper,notree,nostyles,%
nomain,noglossaryindex}

```

Now define `\setupglossaries`:

```

\newcommand*{\setupglossaries}[1]{%
  \renewcommand*{\@gls@setacrstyle}{}%
  \ifglsacrshortcuts
    \def\@gls@setupshortcuts{\glsacrshortcutstrue}%
  \else
    \def\@gls@setupshortcuts{%
      \ifglsacrshortcuts
        \DefineAcronymSynonyms
      \fi
    }%
  \fi
  \glsacrshortcutsfalse
  \let\@gls@do@numbersdef\relax
  \let\@gls@do@symbolssdef\relax
  \let\@gls@do@indexdef\relax
  \let\@gls@do@acronymsdef\relax
  \ifglsentrycounter
    \let\@gls@doentrycounterdef\relax
  \else
    \let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
  \fi
  \ifglsesubentrycounter
    \let\@gls@dosubentrycounterdef\relax
  \else
    \let\@gls@dosubentrycounterdef\@gls@define@glossarysubentrycounter
  \fi
  \setkeys{glossaries.sty}{#1}%
  \@gls@setacrstyle
  \@gls@setupshortcuts
}

```

```

\@gls@do@acronymsdef
\@gls@do@numbersdef
\@gls@do@symbolssdef
\@gls@do@indexdef
\@gls@doentrycounterdef
\@gls@dosubentrycounterdef
}

```

If chapters are defined and the user has requested the section counter as a package option, `\@chapter` will be modified so that it adds a `section.<n>.0` target, otherwise entries placed before the first section of a chapter will have undefined links.

The same problem will also occur if a lower sectional unit is used, but this is less likely to happen. If it does, or if you change `\glscounter` to `section` later, you will have to specify a different counter for the entries that give rise to a `name{<section-level>.<n>.0}` non-existent warning (e.g. `\gls[counter=chapter]{label}`).

```

\ifthenelse{\equal{\glscounter}{section}}%
{%
  \ifcsundef{chapter}{}%
  {%
    \let\@gls@old@chapter\@chapter
    \def\@chapter[#1]#2{\@gls@old@chapter[#1]#2}%
    \ifcsundef{hyperdef}{\hyperdef{section}{\thesection}{}}%
  }%
}%
{}

```

`\@gls@onlypremakeg` Some commands only have an effect when used before `\makeglossaries`. So define a list of commands that should be disabled after `\makeglossaries`

```

\newcommand*{\@gls@onlypremakeg}{}

```

`\@onlypremakeg` Adds the specified control sequence to the list of commands that must be disabled after `\makeglossaries`.

```

\newcommand*{\@onlypremakeg}[1]{%
  \ifx\@gls@onlypremakeg\@empty
    \def\@gls@onlypremakeg{#1}%
  \else
    \expandafter\toks@\expandafter{\@gls@onlypremakeg}%
    \edef\@gls@onlypremakeg{\the\toks@,\noexpand#1}%
  \fi
}

```

`\@disable@onlypremakeg` Disable all commands listed in `\@gls@onlypremakeg`

```

\newcommand*{\@disable@onlypremakeg}{%
\@for\@thiscs:=\@gls@onlypremakeg\do{%
  \expandafter\@disable@premakecs\@thiscs%
}}

```

`\@disable@premakecs` Disables the given command.

```
\newcommand*{\@disable@premakecs}[1]{%
  \def#1{\PackageError{glossaries}{\string#1\space may only be
    used before \string\makeglossaries}{You can't use
    \string#1\space after \string\makeglossaries}}%
}
```

2.3 Predefined Text

Set up default textual tags that are used by this package. Some of the names may already be defined (e.g. by `babel`) so `\providecommand` is used.

Main glossary title:

```
\glossaryname
\providecommand*\glossaryname{Glossary}
```

The title for the `acronym` glossary type (which is defined if `acronym` package option is used) is given by `\acronymname`. If the `acronym` package option is not used, `\acronymname` won't be used.

```
\acronymname
\providecommand*\acronymname{Acronyms}
```

```
\glstocctitle Sets the TOC title for the given glossary.
\newcommand*\glstocctitle[1]{%
  \def\glossarytocctitle{\csname @glotype@#1@title\endcsname}}
```

The following commands provide text for the headers used by some of the tabular-like glossary styles. Whether or not they get used in the glossary depends on the glossary style.

```
\entryname
\providecommand*\entryname{Notation}
```

```
\descriptionname
\providecommand*\descriptionname{Description}
```

```
\symbolname
\providecommand*\symbolname{Symbol}
```

```
\pagelistname
\providecommand*\pagelistname{Page List}
```

Labels for `makeindex`'s symbol and number groups:

```
\glssymbolsgroupname
\providecommand*\glssymbolsgroupname{Symbols}
```

```

\glsnumbersgroupname
    \providecommand*{\glsnumbersgroupname}{Numbers}

\glspluralsuffix The default plural is formed by appending \glspluralsuffix to the singular
form.
    \newcommand*{\glspluralsuffix}{s}

\glsacrpluralsuffix Default plural suffix for acronyms
    \newcommand*{\glsacrpluralsuffix}{\glspluralsuffix}

\glsupacrpluralsuffix
    \newcommand*{\glsupacrpluralsuffix}{\glstextup{\glsacrpluralsuffix}}

\seename
    \providecommand*{\seename}{see}

\andname
    \providecommand*{\andname}{\&}

Add multi-lingual support. Thanks to everyone who contributed to the trans-
lations from both comp.text.tex and via email.

\RequireGlossariesLang
    \newcommand*{\RequireGlossariesLang}[1]{%
        \ifundefined{ver@glossaries-#1.ldf}{\input{glossaries-#1.ldf}}{}%
    }

\ProvidesGlossariesLang
    \newcommand*{\ProvidesGlossariesLang}[1]{%
        \ProvidesFile{glossaries-#1.ldf}%
    }

\addglossarytocaptions Does nothing if translator hasn't been loaded.
    \newcommand*{\addglossarytocaptions}[1]{}

    As from v4.12, multilingual support has been split off into independently-
    maintained language modules.

    \ifglstranslate
    Load tracklang:
        \RequirePackage{tracklang}[2019/11/30]
    Track any languages explicitly requested via the languages option.
        \ifdefvoid\@gls@requested@languages{%
            {%
                \for\@gls@thislang:=\@gls@requested@languages\do{%
                    \TrackIfKnownLanguage{\@gls@thislang}{}%
                    {\GlossariesWarning{unknown language ‘\@gls@thislang’}}%
                }%
            }%
        }%

```


Load translator if required.

```
\@gls@usetranslator
```

If using `translator`, `\glossaryname` should be defined in terms of `\translate`, but if `babel` is also loaded, it will redefine `\glossaryname` whenever the language is set, so override it. (Don't use `\addto` as `polyglossia` doesn't define it.)

```
\@ifpackageloaded{translator}
{%
```

If the language options have been specified through the document class, then `translator` can pick them up. If not, `translator` will default to English and any language option passed to `babel` won't be detected, so if `\trans@languages` is just `English` and `\bbl@loaded` isn't simply `english`, then don't use the `translator` dictionaries.

```
\ifboolexpr
{
  test {\ifdefstring{\trans@languages}{English}}
  and not
  test {\ifdefstring{bbl@loaded}{english}}
}
{%
  \let\glsifusetranslator\@secondoftwo
}%
{%
  \usedictionary{glossaries-dictionary}%
  \renewcommand*{\addglossarytocaptions}[1]{%
    \ifcsundef{captions#1}{}%
    {%
      \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
      \expandafter\toks@\expandafter{\@gls@tmp
        \renewcommand*{\glossaryname}{\translate{Glossary}}}%
    }%
    \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
  }%
}%
}%
}%
{}%
```

Check for tracked languages

```
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
  }%
  \RequireGlossariesLang{\CurrentTrackedTag}%
}%
{}%
```

```

        \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
} %
} %
{} %

```

if using translator use translator interface.

```

\glsifusetranslator
{%
  \renewcommand*\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*\glossaryname{\translate{Glossary}}%
  \renewcommand*\acronymname{\translate{Acronyms}}%
  \renewcommand*\entryname{\translate{Notation (glossaries)}}%
  \renewcommand*\descriptionname{%
    \translate{Description (glossaries)}}%
  \renewcommand*\symbolname{\translate{Symbol (glossaries)}}%
  \renewcommand*\pagelistname{%
    \translate{Page List (glossaries)}}%
  \renewcommand*\glsymbolsgroupname{%
    \translate{Symbols (glossaries)}}%
  \renewcommand*\glsnumbersgroupname{%
    \translate{Numbers (glossaries)}}%
}{}%
\fi

```

Load datatool-base after languages tracked in case there's support for datatool-base as well. Note that datatool-base is always required, regardless of localisation, but datatool-base v3.0+ will also load tracklang if not already loaded.

```
\RequirePackage{datatool-base}
```

`\nopostdesc` Provide a means to suppress description terminator for a given entry. (Useful for entries with no description.) Has no effect outside the glossaries.

```
\DeclareRobustCommand*\nopostdesc{}
```

`\@nopostdesc` Suppress next description terminator.

```

\newcommand*\@nopostdesc{%
  \let\org@glspostdescription\glspostdescription
  \def\glspostdescription{%
    \let\glspostdescription\org@glspostdescription}%
}

```

`\@no@post@desc` Used for comparison purposes.

```
\newcommand*\@no@post@desc}{\nopostdesc}
```

`\glspar` Provide means of having a paragraph break in glossary entries

```
\newcommand{\glspar}{\par}
```

`\setStyleFile` Sets the style file. The relevant extension is appended.

```
\newcommand{\setStyleFile}[1]{%
  \renewcommand*\gls@istfilebase}{#1}%
```

Just in case `\istfilename` has been modified.

```
\ifglxindy
  \def\istfilename{\gls@istfilebase.xdy}
\else
  \def\istfilename{\gls@istfilebase.ist}
\fi
}
```

This command only has an effect prior to using `\makeglossaries`.

```
\@onlypremakeg\setStyleFile
```

The name of the `makeindex` or `xindy` style file is given by `\istfilename`. This file is created by `\writeist` (which is used by `\makeglossaries`) so redefining this command will only have an effect if it is done *before* `\makeglossaries`. As from v1.17, use `\setStyleFile` instead of directly redefining `\istfilename`.

`\istfilename`

```
\ifglxindy
  \def\istfilename{\gls@istfilebase.xdy}
\else
  \def\istfilename{\gls@istfilebase.ist}
\fi
```

`\gls@istfilebase`

```
\newcommand*\gls@istfilebase}{\jobname}
```

The `makeglossaries` Perl script picks up this name from the auxiliary file. If the name ends with `.xdy` it calls `xindy` otherwise it calls `makeindex`. Since its not required by \LaTeX , `\@istfilename` ignores its argument.

`\@istfilename`

```
\newcommand*\@istfilename}[1]{} 
```

This command is the value of the `page_compositor` `makeindex` key. Again, any redefinition of this command must take place *before* `\writeist` otherwise it will have no effect. As from 1.17, use `\glsSetCompositor` instead of directly redefining `\glscompositor`.

`\glscompositor`

```
\newcommand*\glscompositor}{.}
```

`\glsSetCompositor` Sets the compositor.

```
\newcommand*\glsSetCompositor[1]{%
  \renewcommand*\glscompositor{#1}}
```

Only use before `\makeglossaries`

```
\@onlypremakeg\glsSetCompositor
```

(The page compositor is usually defined as a dash when using `makeindex`, but most of the standard counters used by L^AT_EX use a full stop as the compositor, which is why I have used it as the default.) If `xindy` is used `\glscompositor` only affects the `arabic-page-numbers` location class.

`\@glsAlphacompositor` This is only used by `xindy`. It specifies the compositor to use when location numbers are in the form `<letter><compositor><number>`. For example, if `\@glsAlphacompositor` is set to “.” then it allows locations such as A.1 whereas if `\@glsAlphacompositor` is set to “-” then it allows locations such as A-1.

```
\newcommand*\@glsAlphacompositor{\glscompositor}
```

`\glsSetAlphaCompositor` Sets the alpha compositor.

```
\ifglsxindy
  \newcommand*\glsSetAlphaCompositor[1]{%
    \renewcommand*\@glsAlphacompositor{#1}}
\else
  \newcommand*\glsSetAlphaCompositor[1]{%
    \glsnoxindywarning\glsSetAlphaCompositor}
\fi
```

Can only be used before `\makeglossaries`

```
\@onlypremakeg\glsSetAlphaCompositor
```

`\gls@suffixF` Suffix to use for a two page list. This overrides the separator and the closing page number if set to something other than an empty macro.

```
\newcommand*\gls@suffixF}{}
```

`\glsSetSuffixF` Sets the suffix to use for a two page list.

```
\newcommand*\glsSetSuffixF[1]{%
  \renewcommand*\gls@suffixF{#1}}
```

Only has an effect when used before `\makeglossaries`

```
\@onlypremakeg\glsSetSuffixF
```

`\gls@suffixFF` Suffix to use for a three page list. This overrides the separator and the closing page number if set to something other than an empty macro.

```
\newcommand*\gls@suffixFF}{}
```

`\glsSetSuffixFF` Sets the suffix to use for a three page list.

```
\newcommand*\glsSetSuffixFF}[1]{%
  \renewcommand*\gls@suffixFF{#1}%
}
```

`\glsnumberformat` The command `\glsnumberformat` indicates the default format for the page numbers in the glossary. (Note that this is not the same as `\glossaryentrynumbers`, but applies to individual numbers or groups of numbers within an entry’s associated number list.) If hyperlinks are defined, it will use `\glshypernumber`, otherwise it will simply display its argument “as is”.

```
\ifcsundef{hyperlink}%
  {%
    \newcommand*\glsnumberformat}[1]{#1}%
  }%
  {%
    \newcommand*\glsnumberformat}[1]{\glshypernumber{#1}}%
  }
```

`\glslocationcstoencap` Used by `makeglossaries` to move commands out of the location numbers into the `encap`. The actual `encap` needs to be closest to the location in the event that hyperlinks are required.

```
\newcommand*\glslocationcstoencap}[3]{\csuse{#2}{\csuse{#1}{#3}}}
```

Individual numbers in an entry’s associated number list are delimited using `\delimN` (which corresponds to the `delim_n makeindex` keyword). The default value is a comma followed by a space.

`\delimN`

```
\newcommand*\delimN}{, }
```

A range of numbers within an entry’s associated number list is delimited using `\delimR` (which corresponds to the `delim_r makeindex` keyword). The default is an en-dash.

`\delimR`

```
\newcommand*\delimR}{--}
```

The glossary preamble is given by `\glossarypreamble`. This will appear after the glossary sectioning command, and before the `theglossary` environment. It is designed to allow the user to add information pertaining to the glossary (e.g. “page numbers in italic indicate the primary definition”) therefore `\glossarypreamble` shouldn’t be affected by the glossary style. (So if you define your own glossary style, don’t have it change `\glossarypreamble`.) The preamble is empty by default. If you have multiple glossaries, and you want a different preamble for each glossary, you will need to use `\printglossary` for each glossary type, instead of `\printglossaries`, and redefine `\glossarypreamble` before each `\printglossary`.

```
\glossary preamble
  \newcommand*{\glossary preamble}{%
    \csuse{@glossary preamble@currentglossary}%
  }
```

```
\setglossary preamble[<type>]{<text>}
```

```
\setglossary preamble
```

Code provided by Michael Pock.

```
\newcommand{\setglossary preamble}[2][\gls default type]{%
  \ifglossary exists*{#1}%
  {\csgdef{@glossary preamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
```

The glossary postamble is given by `\glossary postamble`. This is provided to allow the user to add something after the end of the `theglossary` environment (again, this shouldn't be affected by the glossary style). It is, of course, possible to simply add the text after `\printglossary`, but if you only want the postamble to appear after the first glossary, but not after subsequent glossaries, you can do something like:

```
%\renewcommand{\glossary postamble}{For a complete list of terms
%see \cite{blah}\gdef\glossary preamble{}}
%
```

```
\glossary postamble
  \newcommand*{\glossary postamble}{}
```

`\glossary section` The sectioning command that starts a glossary is given by `\glossary section`. (This does not form part of the glossary style, and so should not be changed by a glossary style.) If `\phantom section` is defined, it uses `\p@glossary section`, otherwise it uses `\@glossary section`.

```
\newcommand*{\glossary section}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantom section}%
  {%
    \@glossary section{#1}{#2}%
  }%
  {%
    \p@glossary section{#1}{#2}%
  }%
  \gls glossary mark{\glossary to c title}%
}
```

`\gls glossary mark` Sets the header mark for the glossary. Takes the glossary short (TOC) title as the argument. If `\glossary mark` hasn't been defined, it will be defined shortly, so use that. Otherwise don't use it as it may not be compatible.

```

\ifcsundef{glossarymark}%
{%
  \newcommand{\glsglossarymark}[1]{\glossarymark{#1}}
}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand{\glsglossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUHead{#1}}{\memUHead{#1}}%
      \else
        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand{\glsglossarymark}[1]{%
      \ifglsucmark
        \@mkboth{\glssupercase{#1}}{\glssupercase{#1}}%
      \else
        \@mkboth{#1}{#1}%
      \fi
    }
  }
}
}

```

`\glossarymark` Provided for backward compatibility:

```

\providecommand{\glossarymark}[1]{%
  \ifglsucmark
    \@mkboth{\glssupercase{#1}}{\glssupercase{#1}}%
  \else
    \@mkboth{#1}{#1}%
  \fi
}

```

The required sectional unit is given by `\@glossarysec` which was defined by the `section` package option. The starred form of the command is chosen. If you don't want any sectional command, you will need to redefine `\glossarysection`. The sectional unit can be changed, if different sectional units are required.

`\setglossarysection`

```

\newcommand*{\setglossarysection}[1]{%
  \setkeys{glossaries.sty}{section=#1}}

```

The command `\glossarysection` indicates how to start the glossary section if `\phantomsection` is not defined.

`\@glossarysection`

```

\newcommand*{\@glossarysection}[2]{%

```

```

\ifdefempty\@glossarysecstar
{%
  \csname\@glossarysec\endcsname[#1]{#2}%
}%
{%
  \csname\@glossarysec\endcsname*{#2}%
  \@gls@toc{#1}{\@glossarysec}%
}%

```

Do automatic labelling if required

```

\@glossaryseclabel
}

```

As `\@glossarysection`, but put in `\phantomsection`, and swap where `\@gls@toc` goes. If using chapters do a `\clearpage`. This ensures that the hyper link from the table of contents leads to the line above the heading, rather than the line below it.

`\@p@glossarysection`

```

\newcommand*\@p@glossarysection}[2]{%
  \gls@clearpage
  \phantomsection
  \ifdefempty\@glossarysecstar
  {%
    \csname\@glossarysec\endcsname{#2}%
  }%
  {%
    \@gls@toc{#1}{\@glossarysec}%
    \csname\@glossarysec\endcsname*{#2}%
  }%
}

```

Do automatic labelling if required

```

\@glossaryseclabel
}

```

`\gls@docclearpage` The `\gls@docclearpage` command is used to issue a `\clearpage` (or `\cleardoublepage`) depending on whether the glossary sectional unit is a chapter. If the sectional unit is something else, do nothing.

```

\newcommand*\gls@docclearpage{%
  \ifthenelse{\equal{\@glossarysec}{chapter}}%
  {%
    \ifcsundef{cleardoublepage}%
    {%
      \clearpage
    }%
  }%
  {\ifcsdef{if@openright}%
  {%
    \if@openright
      \cleardoublepage
    }%
  }%
}

```



```

        \else
        \clearpage
        \fi
    }%
    {%
    \cleardoublepage
    }%
    }%
    }%
    {}%
}

```

`\glsclearpage` This just calls `\gls@docclearpage`, but it makes it easier to have a user command so that the user can override it.

```
\newcommand*\glsclearpage{\gls@docclearpage}
```

The glossary is added to the table of contents if `glstoc` flag set. If it is set, `\@gls@toc` will add a line to the `.toc` file, otherwise it will do nothing. (The first argument to `\@gls@toc` is the title for the table of contents, the second argument is the sectioning type.)

```

\@gls@toc
\newcommand*\@gls@toc[2]{%
  \ifglstoc
  \ifglsnumberline
  \addcontentsline{toc}{#2}{\protect\numberline{#1}}%
  \else
  \addcontentsline{toc}{#2}{#1}%
  \fi
\fi
}

```

2.4 Xindy

This section defines commands that only have an effect if `xindy` is used to sort the glossaries.

`\glsnoxindywarning` Issues a warning if `xindy` hasn't been specified. These warnings can be suppressed by redefining `\glsnoxindywarning` to ignore its argument

```

\newcommand*\glsnoxindywarning[1]{%
  \GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
}

```

`\glsnomakeindexwarning` Reverse for commands that may only be used with `makeindex`.

```

\newcommand*\glsnomakeindexwarning[1]{%
  \GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%
}

```

```

\@xdyattributes Define list of attributes (\string is used in case the double quote character has
been made active)
\ifglxindy
  \edef\@xdyattributes{\string"default\string"}%
\fi

\@xdyattributelist Comma-separated list of attributes.
\ifglxindy
  \edef\@xdyattributelist{}%
\fi

\@xdylocref Define list of markup location references.
\ifglxindy
  \def\@xdylocref{}
\fi

\@gls@ifinlist
\newcommand*\@gls@ifinlist[4]{%
  \def\@do@ifinlist##1,#1,##2\end@do@ifinlist{%
    \def\@gls@listsuffix{##2}%
    \ifx\@gls@listsuffix\@empty
      #4%
    \else
      #3%
    \fi
  }%
  \@do@ifinlist,#2,#1,\end@do@ifinlist
}

\GlsAddXdyCounters Need to know all the counters that will be used in location numbers for Xindy.
Argument may be a single counter name or a comma-separated list of counter
names.
\ifglxindy
  \newcommand*\@xdycounters{\@glscounter}
  \newcommand*\GlsAddXdyCounters[1]{%
    \@for\@gls@ctr:=#1\do{%
      Check if already in list before adding.
      \edef\@do@addcounter{%
        \noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
        {%
          \noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
            \noexpand\@gls@ctr}%
        }%
      }%
      \@do@addcounter
    }
  }

```

Only has an effect before `\writeist`:

```
\@onlypremakeg\GlsAddXdyCounters
\else
\newcommand*\GlsAddXdyCounters[1]{%
\glsnoxindywarning\GlsAddXdyAttribute
}
\fi
```

`\@disabled@glsaddxdycounters` Counters must all be identified before adding attributes.

```
\newcommand*\@disabled@glsaddxdycounters{%
\PackageError{glossaries}{\string\GlsAddXdyCounters\space
can't be used after \string\GlsAddXdyAttribute}{Move all
occurrences of \string\GlsAddXdyCounters\space before the first
instance of \string\GlsAddXdyAttribute}%
}
```

`\GlsAddXdyAttribute` Adds an attribute.

```
\ifglsxindy
```

First define internal command that adds an attribute for a given counter (2nd argument is the counter):

```
\newcommand*\@glsaddxdyattribute[2]{%
```

Add to xindy attribute list

```
\edef\xdyattributes{\@xdyattributes ^^J
\string"#2#1\string"}%
```

Add to xindy markup location.

```
\expandafter\toks@\expandafter{\@xdylocref}%
\edef\xdylocref{\the\toks@ ^^J%
(markup-locref
:open \string"\glstildechar n%
\expandafter\string\csname glsX#2X#1\endcsname
\string" ^^J
:close \string"\string" ^^J
:attr \string"#2#1\string")}%
```

Define associated attribute command `\glsX<counter>X<attribute>{\<Hprefix>}{<n>}`

```
\expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
\setentrycounter[##1]{#2}\csname #1\endcsname{##2}%
}%
}
```

High-level command:

```
\newcommand*\GlsAddXdyAttribute[1]{%
```

Add to comma-separated attribute list

```
\ifx\xdyattributelist\@empty
\edef\xdyattributelist{#1}%
\else
```

```

\edef\@xdyattributelist{\@xdyattributelist,#1}%
\fi

```

First add the attribute without the counter name:

```

\edef\@xdyattributes{\@xdyattributes ^^J
\string"#1\string"}%

```

Iterate through all specified counters and add counter-dependent attributes:

```

\@for\@this@counter:=\@xdycounters\do{%
\protected@edef\gls@do@addxdyattribute{%
\noexpand\@glsaddxdyattribute{#1}{\@this@counter}%
}
\gls@do@addxdyattribute
}%

```

All occurrences of `\GlsAddXdyCounters` must be used before this command

```

\let\GlsAddXdyCounters\@disabled@glsaddxdycounters
}

```

Only has an effect before `\writeist`:

```

\@onlypremakeg\GlsAddXdyAttribute
\else
\newcommand*\GlsAddXdyAttribute[1]{%
\glsnoxindywarning\GlsAddXdyAttribute}
\fi

```

`\gls@addpredefinedattributes` Add known attributes for all defined counters

```

\ifglsxindy
\newcommand*{\@gls@addpredefinedattributes}{%
\GlsAddXdyAttribute{glsnumberformat}
\GlsAddXdyAttribute{textrm}
\GlsAddXdyAttribute{textsf}
\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glsnumber}
\GlsAddXdyAttribute{hyperm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hypertt}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}
\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}
}

```

```

    \GlsAddXdyAttribute{glsignore}
  }
  \else
    \let\@gls@addpredefinedattributes\relax
  \fi

```

`\@xdyuseralphabets` List of additional alphabets

```
\def\@xdyuseralphabets{}
```

`\GlsAddXdyAlphabet` `\GlsAddXdyAlphabet{<name>}{<definition>}` adds a new alphabet called `<name>`. The definition must use xindy syntax.

```

\ifglsxindy
  \newcommand*\GlsAddXdyAlphabet}[2]{%
  \edef\@xdyuseralphabets{%
    \@xdyuseralphabets ^^J
    (define-alphabet "#1" (#2))}
  \else
  \newcommand*\GlsAddXdyAlphabet}[2]{%
    \glsnoxindywarning\GlsAddXdyAlphabet}
  \fi

```

This code is only required for xindy:

```
\ifglsxindy
```

`\@gls@xdy@locationlist` List of predefined location names.

```

\newcommand*\@gls@xdy@locationlist{%
  roman-page-numbers,%
  Roman-page-numbers,%
  arabic-page-numbers,%
  alpha-page-numbers,%
  Alpha-page-numbers,%
  Appendix-page-numbers,%
  arabic-section-numbers%
}

```

Each location class `<name>` has the format stored in `\@gls@xdy@Lclass@<name>`. Set up predefined formats.

`\@Lclass@roman-page-numbers` Lower case Roman numerals (i, ii, ...). In the event that `\roman` has been redefined to produce a fancy form of roman numerals, attempt to work out how it will be written to the output file.

```

\protected@edef\@gls@roman{\@roman{0}\string"
  \string"roman-numbers-lowercase\string" :sep \string"}%
\@onelevel@sanitize\@gls@roman
\edef\@gls@tmp{\string" \string"roman-numbers-lowercase\string"
  :sep \string"}%
\@onelevel@sanitize\@gls@tmp
\ifx\@gls@tmp\@gls@roman
  \expandafter

```

```

\edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
  \string"roman-numbers-lowercase\string"%
}%
\else
\expandafter
\edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{
:sep \string"\@gls@roman\string"%
}%
\fi

\y@Lclass@Roman-page-numbers Upper case Roman numerals (I, II, ...).
\expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
\string"roman-numbers-uppercase\string"%
}%

\@Lclass@arabic-page-numbers Arabic numbers (1, 2, ...).
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
\string"arabic-numbers\string"%
}%

\y@Lclass@alpha-page-numbers Lower case alphabetical (a, b, ...).
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
\string"alpha\string"%
}%

\y@Lclass@Alpha-page-numbers Upper case alphabetical (A, B, ...).
\expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
\string"ALPHA\string"%
}%

\class@Appendix-page-numbers Appendix style locations (e.g. A-1, A-2, ..., B-1, B-2, ...). The separator is
given by \@glsAlphacompositor.
\expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
\string"ALPHA\string"
:sep \string"\@glsAlphacompositor\string"
\string"arabic-numbers\string"%
}

\class@arabic-section-numbers Section number style locations (e.g. 1.1, 1.2, ...). The compositor is given by
\glscompositor.
\expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
\string"arabic-numbers\string"
:sep \string"\glscompositor\string"
\string"arabic-numbers\string"%
}%

\@xdyuserlocationdefs List of additional location definitions (separated by ^^J)
\def\@xdyuserlocationdefs{}

```

`\@xdyuserlocationnames` List of additional user location names

```
\def\@xdyuserlocationnames{}
```

End of xindy-only block:

```
\fi
```

`\@xdycrossrefhook` Hook used after writing cross-reference class information.

```
\ifglxindy
\newcommand\@xdycrossrefhook{}
\fi
```

`\GlsAddXdyLocation` `\GlsAddXdyLocation` [*prefix-loc*] {*name*} {*definition*} Define a new location called *name*. The definition must use xindy syntax. (Note that this doesn't check to see if the location is already defined. That is left to xindy to complain about.)

```
\ifglxindy
\newcommand*\GlsAddXdyLocation}[3][{}]{%
\def\@gls@tmp{#1}%
\ifx\@gls@tmp\@empty
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep \string"{}\glsopenbrace\string" #3
:sep \string"\glsclosebrace\string"))
}%
\else
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep "\glsopenbrace"
#1
:sep "\glsclosebrace\glsopenbrace" #3
:sep "\glsclosebrace"))
}%
\fi

\edef\@xdyuserlocationnames{%
\@xdyuserlocationnames^^J\space\space\space
\string"#2\string"}%
}
```

Only has an effect before `\writeist`:

```
\@onlypremakeg\GlsAddXdyLocation
\else
\newcommand*\GlsAddXdyLocation}[2]{%
\glsnoxindywarning\GlsAddXdyLocation}
\fi
```

`\@xdylocationclassorder` Define location class order

```

\ifglxindy
\def\@xdylocationclassorder{^^J\space\space\space
\string"roman-page-numbers\string"^^J\space\space\space
\string"arabic-page-numbers\string"^^J\space\space\space
\string"arabic-section-numbers\string"^^J\space\space\space
\string"alpha-page-numbers\string"^^J\space\space\space
\string"Roman-page-numbers\string"^^J\space\space\space
\string"Alpha-page-numbers\string"^^J\space\space\space
\string"Appendix-page-numbers\string"
\@xdyuserlocationnames^^J\space\space\space
\string"see\string"
}
\fi

```

Change the location order.

`\GlsSetXdyLocationClassOrder`

```

\ifglxindy
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\def\@xdylocationclassorder{#1}}
\else
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\glsnoxywarning\GlsSetXdyLocationClassOrder}
\fi

```

`\@xdysortrules` Define sort rules

```

\ifglxindy
\def\@xdysortrules{}
\fi

```

`\GlsAddSortRule` Add a sort rule

```

\ifglxindy
\newcommand*\GlsAddSortRule[2]{%
\expandafter\toks@\expandafter{\@xdysortrules}%
\protected@edef\@xdysortrules{\the\toks@ ^^J
(sort-rule \string"#1\string" \string"#2\string")}%
}
\else
\newcommand*\GlsAddSortRule[2]{%
\glsnoxywarning\GlsAddSortRule}
\fi

```

`\@xdyrequiredstyles` Define list of required styles (this should be a comma-separated list of xindy styles)

```

\ifglxindy
\def\@xdyrequiredstyles{tex}
\fi

```

`\GlsAddXdyStyle` Add a xindy style to the list of required styles


```

\ifglxindy
  \newcommand*\GlsAddXdyStyle[1]{%
    \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
  \newcommand*\GlsAddXdyStyle[1]{%
    \glsnoxindywarning\GlsAddXdyStyle}
\fi

```

`\GlsSetXdyStyles` Reset the list of required styles

```

\ifglxindy
  \newcommand*\GlsSetXdyStyles[1]{%
    \edef\@xdyrequiredstyles{#1}}
\else
  \newcommand*\GlsSetXdyStyles[1]{%
    \glsnoxindywarning\GlsSetXdyStyles}
\fi

```

`\findrootlanguage` The command `\findrootlanguage` was used to determine the root language, using a bit of trickery. This has been deferred to `tracklang`. Removed in v4.50.

`\@xdylanguage` The xindy language setting is required by `makeglossaries`, so provide a command for `makeglossaries` to pick up the information from the auxiliary file. This command is not needed by the `glossaries` package, so define it to ignore its arguments.

```

\def\@xdylanguage#1#2{}

```

`\GlsSetXdyLanguage` Define a command that allows the user to set the language for a given glossary type. The first argument indicates the glossary type. If omitted the main glossary is assumed. This uses the unstarred form of `\ifglossaryexists` because ignored glossaries can't be used with xindy.

```

\ifglxindy
  \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
    \ifglossaryexists{#1}{%
      \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
    }{%
      \PackageError{glossaries}{Can't set language type for
        glossary type '#1' --- no such glossary}{%
        You have specified a glossary type that doesn't exist}}
\else
  \newcommand*\GlsSetXdyLanguage[2][]{%
    \glsnoxindywarning\GlsSetXdyLanguage}
\fi

```

`\@gls@codepage` The xindy codepage setting is required by `makeglossaries`, so provide a command for `makeglossaries` to pick up the information from the auxiliary file. This command is not needed by the `glossaries` package, so define it to ignore its arguments.

```

\def\@gls@codepage#1#2{}

```

`\GlsSetXdyCodePage` Define command to set the code page.

```
\ifglxindy
  \newcommand*\GlsSetXdyCodePage}[1]{%
    \renewcommand*\gls@codepage}{#1}%
  }
```

Suggested by egreg:

```
\AtBeginDocument{%
  \ifx\gls@codepage\@empty
    \ifpackageloaded{fontspec}{\def\gls@codepage{utf8}}{}%
  \fi
}
\else
  \newcommand*\GlsSetXdyCodePage}[1]{%
    \glsnoxywarning\GlsSetXdyCodePage}
\fi
```

`\@xdylettergroups` Store letter group definitions.

```
\ifglxindy
  \ifglxindy@glsnumbers
    \def\@xdylettergroups{(define-letter-group
      \string\glsnumbers\string^^J\space\space\space
      :prefixes (\string"0\string" \string"1\string"
        \string"2\string" \string"3\string" \string"4\string"
        \string"5\string" \string"6\string" \string"7\string"
        \string"8\string" \string"9\string")^^J\space\space\space
      \@xdynumbergrouporder)}
  \else
    \def\@xdylettergroups{}
  \fi
\fi
```

`\GlsAddLetterGroup` Add a new letter group. The first argument is the name of the letter group. The second argument is the xindy code specifying prefixes and ordering.

```
\newcommand*\GlsAddLetterGroup[2]{%
  \expandafter\toks@\expandafter{\@xdylettergroups}%
  \protected@edef\@xdylettergroups{the\toks@^^J%
    (define-letter-group \string"#1\string"^^J\space\space\space#2)}%
  }%
```

2.5 Loops and conditionals

`\forallglossaries` To iterate through all glossaries (or comma-separated list of glossary names given in optional argument) use:

```
\forallglossaries[glossary list]{cmd}{code}
```

where `<cmd>` is a control sequence which will be set to the name of the glossary in the current iteration.

```
\newcommand*\forallglossaries}[3][\@glo@types]{%
  \@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%
}
```

`\forallacronyms`

```
\newcommand*\forallacronyms}[2]{%
  \@for#1:=\@glsacronymlists\do{\ifx#1\@empty\else#2\fi}%
}
```

`\forallglsentries` To iterate through all entries in a given glossary use:

```
\forallglsentries[<type>]{<cmd>}{<code>}
```

where *<type>* is the glossary label and *<cmd>* is a control sequence which will be set to the entry label in the current iteration.

```
\newcommand*\forallglsentries}[3][\glsdefaulttype]{%
  \protected@edef\@glo@list{\csname glist@#1\endcsname}%
  \@for#2:=\@glo@list\do
  {%
    \ifdefempty{#2}{#3}%
  }%
}
```

`\forallglsentries` To iterate through all glossary entries over all glossaries listed in the optional argument (the default is all glossaries) use:

```
\forallglsentries[<glossary list>]{<cmd>}{<code>}
```

Within `\forallglsentries`, the current glossary type is given by `\@this@glo@`.

```
\newcommand*\forallglsentries}[3][\@glo@types]{%
  \expandafter\forallglossaries\expandafter[#1]{\@this@glo@}%
  {%
    \forallglsentries[\@this@glo@]{#2}{#3}%
  }%
}
```

`\ifglossaryexists` To check to see if a glossary exists use:

```
\ifglossaryexists{<type>}{<true-text>}{<false-text>}
```

where *<type>* is the glossary's label. The unstarred form will do *<false-text>* for ignored glossaries. The starred form will do *<true-text>* for ignored glossaries.

```
\newcommand*\ifglossaryexists{%
  \@ifstar\s@ifglossaryexists\@ifglossaryexists
}
```

`\@ifglossaryexists` Unstarred form only tests the existence of non-ignored glossaries.

```
\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{@glotype@#1@out}{#3}{#2}%
}
```

`\s@ifglossaryexists` Starred form includes ignored glossaries.

```
\newcommand{\s@ifglossaryexists}[3]{%
  \ifcsundef{glolist@#1}{#3}{#2}%
}
```

Since the label is used to form the name of control sequences, by default UTF8 etc characters can't be used in the label. A possible workaround is to use `\scantokens`, but commands such as `\glentrytext` will no longer be usable in sectioning, caption etc commands. If the user really wants to be able to construct a label with UTF8 characters, allow them the means to do so (but on their own head be it, if they then use entries in `\section` etc). This can be done via:

```
%\renewcommand*{\glsdetoklabel}[1]{\scantokens{#1\noexpand}}
%
```

(Note, don't use `\detokenize` or it will cause commands like `\glsaddall` to fail.) Since redefining `\glsdetoklabel` can cause things to go badly wrong, I'm not going to mention it in the main user guide. Only advanced users who know what they're doing ought to attempt it.

`\glsdetoklabel`

```
\newcommand*{\glsdetoklabel}[1]{#1}
```

`\ifglentryexists` To check to see if a glossary entry has been defined use:

```
\ifglentryexists{<label>}{<true text>}{<>false text>}
```

where *<label>* is the entry's label.

```
\newcommand{\ifglentryexists}[3]{%
  \ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}
```

`\ifglsused` To determine if given glossary entry has been used in the document text yet use:

```
\ifglsused{<label>}{<true text>}{<>false text>}
```

where *<label>* is the entry's label. If true it will do *<true text>* otherwise it will do *<>false text>*.

```
\newcommand*{\ifglsused}[3]{%
  \ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}
```

The following two commands will cause an error if the given condition fails:

`\glsdoifexists` `\glsdoifexists{<label>}{<code>}`

Generate an error if entry specified by *<label>* doesn't exist, otherwise do *<code>*.

```
\newcommand{\glsdoifexists}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}%
  }
```

`\glsdoifnoexists` `\glsdoifnoexists{<label>}{<code>}`

The opposite: only do second argument if the entry doesn't exist. Generate an error message if it exists.

```
\newcommand{\glsdoifnoexists}[2]{%
  \ifglsentryexists{#1}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
      been defined}{#2}%
  }
```

`\glsdoifexistsorwarn` `\glsdoifexistsorwarn{<label>}{<code>}`

Generate a warning if entry specified by *<label>* doesn't exist, otherwise do *<code>*.

```
\newcommand{\glsdoifexistsorwarn}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}%
  }
```

`\glsdoifexistsordo` `\glsdoifexistsordo{<label>}{<code>}{<undef code>}`

Generate an error and do *<undef code>* if entry specified by *<label>* doesn't exist, otherwise do *<code>*.

```
\newcommand{\glsdoifexistsordo}[3]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}%
    #3%
  }
```

`\doifglossarynoexistsordo`

```
\doifglossarynoexistsordo{<label>}{<code>}{<else code>}
```

If glossary given by *<label>* doesn't exist do *<code>* otherwise generate an error and do *<else code>*.

```
\newcommand{\doifglossarynoexistsordo}[3]{%
  \ifglossaryexists*{#1}%
  {%
    \PackageError{glossaries}{Glossary type '#1' already exists}{}%
    #3%
  }%
  {#2}%
}
```

`\ifglshaschildren`

```
\ifglshaschildren{<label>}{<true part>}{<false part>}
```

This is inefficient as it has to search through all entries to find out which ones have the given entry as its parent. It's much easier to use `bib2gls` and get it to store the list of children that have been indexed (which is likely to be more useful).

```
\newrobustcmd{\ifglshaschildren}[3]{%
  \glsdoifexists{#1}%
  {%
    \def\do@glshaschildren{#3}%

    \protected@edef\@gls@thislabel{\glsdetoklabel{#1}}%
    \expandafter\for@gl@entries\expandafter
    [\csname glo@\@gls@thislabel @type\endcsname]
    {\glo@label}%
    {%
      \letcs@glo@parent{glo@\glo@label @parent}%
      \ifdefequal\@gls@thislabel\glo@parent
      {%
        \def\do@glshaschildren{#2}%
        \@endfortrue
      }%
    }%
  }%
  \do@glshaschildren
}
```

`\ifglshasparent`

```
\ifglshasparent{<label>}{<true part>}{<false part>}
```

```
\newcommand{\ifglshasparent}[3]{%
```

```

\glsdoifexists{#1}%
{%
  \ifcsemtyp{glo@glsdetoklabel{#1}@parent}{#3}{#2}%
}%
}

\ifglshasdesc \ifglshasdesc{<label>}{<true part>}{<false part>}
\newcommand*{\ifglshasdesc}[3]{%
  \ifcsemtyp{glo@glsdetoklabel{#1}@desc}%
  {#3}%
  {#2}%
}

\ifglsdescsuppressed \ifglsdescsuppressed{<label>}{<true part>}{<false part>} Does <true part> if
the description is just \nopostdesc otherwise does <false part>.
\newcommand*{\ifglsdescsuppressed}[3]{%
  \ifcsequal{glo@glsdetoklabel{#1}@desc}{@no@post@desc}%
  {#2}%
  {#3}%
}

\ifglshassymbol \ifglshassymbol{<label>}{<true part>}{<false part>}
\newrobustcmd*{\ifglshassymbol}[3]{%
  \letcs{\@glo@symbol}{glo@glsdetoklabel{#1}@symbol}%
  \ifdefempty\@glo@symbol
  {#3}%
  {%
    \ifdefequal\@glo@symbol\@gls@default@value
    {#3}%
    {#2}%
  }%
}

\ifglshaslong \ifglshaslong{<label>}{<true part>}{<false part>}
\newrobustcmd*{\ifglshaslong}[3]{%
  \letcs{\@glo@long}{glo@glsdetoklabel{#1}@long}%
  \ifdefempty\@glo@long
  {#3}%
  {%
    \ifdefequal\@glo@long\@gls@default@value
    {#3}%
    {#2}%
  }%
}

\ifglshasshort \ifglshasshort{<label>}{<true part>}{<false part>}
\newrobustcmd*{\ifglshasshort}[3]{%
  \letcs{\@glo@short}{glo@glsdetoklabel{#1}@short}%
  \ifdefempty\@glo@short

```

```

    {#3}%
    {%
      \ifdefequal\@glo@short\@gls@default@value
        {#3}%
        {#2}%
      }%
    }
  }

```

```

\ifglsfieldvoid{<field>}{<label>}{<true part>}{<false
part>}

```

\ifglsfieldvoid

Provide an expandable way of testing if a field is undefined or empty.

```

\newcommand{\ifglsfieldvoid}[4]{\ifcvoid{glo@glsetoklabel{#2}@#1}{#3}{#4}}

```

```

\ifglsfield{<field>}{<label>}{<true part>}{<false
part>}

```

\ifglsfield

```

\newrobustcmd*{\ifglsfield}[4]{%
  \glsdoifexists{#2}%
  {%
    \letcs{\@glo@thisvalue}{glo@glsetoklabel{#2}@#1}%

```

First check supplied field label is defined.

```

  \ifdef\@glo@thisvalue
    {%

```

Is defined, so now check if empty.

```

      \ifdefempty\@glo@thisvalue
        {%

```

Is empty, so doesn't have field set.

```

          #4%
        }%
      }%
    }%

```

Not empty, so check if set to \@gls@default@value

```

      \ifdefequal\@glo@thisvalue\@gls@default@value
        {%

```

Value is set to the default value.

```

          #4%
        }%
      }%
    }%

```

Non-empty, non-default value. Allow user to access this value through \glscurrentfieldvalue.

```

      \let\glscurrentfieldvalue\@glo@thisvalue

```



```

        #3%
      }%
    }%
  }%
  {%

```

Field given isn't defined, so check if mapping exists.

```

    \@gls@fetchfield{\@gls@thisfield}{#1}%

```

If \@gls@thisfield is defined, we've found a map. If not, the field supplied doesn't exist.

```

    \ifdef\@gls@thisfield
    {%

```

Is defined, so now check if empty.

```

        \letcs{\@glo@thisvalue}{glo@glsdetoklabel{#2}@@gls@thisfield}%
        \ifdefempty\@glo@thisvalue
        {%

```

Is empty so field hasn't been set.

```

        #4%
      }%
    {%

```

Isn't empty so check if it's been set to \@gls@default@value.

```

        \ifdefequal\@glo@thisvalue\@gls@default@value
        {%

```

Value is set to the default value.

```

        #4%
      }%
    {%

```

Non-empty, non-default value. Allow user to access this value through \glscurrentfieldvalue.

```

        \let\glscurrentfieldvalue\@glo@thisvalue
        #3%
      }%
    }%
  }%
  {%

```

Not defined.

```

        \GlossariesWarning{Unknown entry field '#1'}%
        #4%
      }%
    }%
  }%
}

```

\glscurrentfieldvalue

```

\newcommand*{\glscurrentfieldvalue}{}

```

2.6 Defining new glossaries

A comma-separated list of glossary names is stored in `\@glo@types`. When a new glossary type is created, its identifying name is added to this list. This is used by commands that iterate through all glossaries (such as `\makeglossaries` and `\printglossaries`).

`\@glo@types`

```
\newcommand*\@glo@types{,}
```

`\@gls@provide@newglossary` If the user removes the glossary package from their document, ensure the next run doesn't throw a load of undefined control sequence errors when the aux file is parsed.

```
\newcommand*\@gls@provide@newglossary{%
  \protected@write\@auxout{}\string\providecommand\string\@newglossary[4]{}}%
```

Only need to do this once.

```
\let\@gls@provide@newglossary\relax
}
```

`\@gls@defaulttype`

```
\newcommand*\@gls@defaulttype{\glsdefaulttype}
```

`\defglsentryfmt` Allow different glossaries to have different display styles.

```
\newcommand*\defglsentryfmt[2][\glsdefaulttype]{%
  \protected@edef\@glo@tmp{#1}%
  \ifx\@glo@tmp\@gls@defaulttype
```

Argument was `\glsdefaulttype` but it hasn't expanded, which probably means `nomain` was used and no alternative glossary has been defined.

```
  \GlossariesWarning{\string\defglsentryfmt\space attempting to set
    a display style for \string\glsdefaulttype\space but there
    doesn't seem to be a default glossary. If you have used
    'nomain', you need to define an alternative glossary}%
```

```
\else
  \csgdef{gls@\@glo@tmp @entryfmt}{#2}%
\fi
}
```

`\gls@doentryfmt`

```
\newcommand*\gls@doentryfmt[1]{\csuse{gls@#1@entryfmt}}
```

`\@gls@forbidtext` As a security precaution, don't allow the user to specify a 'tex' extension for any of the glossary files. (Just in case a seriously confused novice user doesn't know what they're doing.) The argument must be a control sequence whose replacement text is the requested extension.

```
\newcommand*\@gls@forbidtext[1]{%
  \ifboolexpr{test {\ifdefstring{#1}{tex}}
    or test {\ifdefstring{#1}{TEX}}}
```

```

{%
  \def#1{nottex}%
  \PackageError{glossaries}%
  {Forbidden '.tex' extension replaced with '.nottex'}%
  {I'm sorry, I can't allow you to do something so reckless.\MessageBreak
  Don't use '.tex' as an extension for a temporary file.}%
}%
{%
}%
}

```

`\gls@gobbleopt` Discard optional argument.

```

\newcommand*\gls@gobbleopt{\new@ifnextchar[{\@gls@gobbleopt}]{}}
\def\@gls@gobbleopt[#1]{ }

```

A new glossary type is defined using `\newglossary`. Syntax:

```

\newglossary[log-ext]{name}[in-ext]{out-ext}
{title}[counter]

```

where *log-ext* is the extension of the `makeindex` transcript file, *in-ext* is the extension of the glossary input file (read in by `\printglossary` and created by `makeindex`), *out-ext* is the extension of the glossary output file which is read in by `makeindex` (lines are written to this file by the `\glossary` command), *title* is the title of the glossary that is used in `\glossarysection` and *counter* is the default counter to be used by entries belonging to this glossary. The `makeglossaries` Perl script reads in the relevant extensions from the auxiliary file, and passes the appropriate file names and switches to `makeindex`.

`\newglossary`

```

\newcommand*\newglossary{\@ifstar\s@newglossary\ns@newglossary}

```

`\s@newglossary` The starred version will construct the extension based on the label.

```

\newcommand*\s@newglossary}[2]{%
  \ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
}

```

`\ns@newglossary` Define the unstarred version.

```

\newcommand*\ns@newglossary}[5][glg]{%
  \doifglossarynoexistsordo{#2}%
  {%

```

Check if default has been set

```

  \ifundef\glsdefaultttype
  {%
    \gdef\glsdefaultttype{#2}%
  }{ }%

```

Add this to the list of glossary types:

```
\toks@{#2}\protected@edef\@glo@types{\@glo@types\the\toks@,}%
```

Define a comma-separated list of labels for this glossary type, so that all the entries for this glossary can be reset with a single command. When a new entry is created, its label is added to this list.

```
\expandafter\gdef\csname glolist@#2\endcsname{,}%
```

Store the file extensions:

```
\expandafter\edef\csname @glotype@#2@log\endcsname{#1}%  
\expandafter\edef\csname @glotype@#2@in\endcsname{#3}%  
\expandafter\edef\csname @glotype@#2@out\endcsname{#4}%  
\expandafter\@gls@forbidtextext\csname @glotype@#2@log\endcsname  
\expandafter\@gls@forbidtextext\csname @glotype@#2@in\endcsname  
\expandafter\@gls@forbidtextext\csname @glotype@#2@out\endcsname
```

Store the title:

```
\expandafter\def\csname @glotype@#2@title\endcsname{#5}%  
  
\@gls@provide@newglossary  
\protected@write\@auxout{\string\@newglossary{#2}{#1}{#3}{#4}}%
```

How to display this entry in the document text (uses `\glsentry` by default). This can be redefined by the user later if required (see `\defglsentry`). This may already have been defined if this has been specified as a list of acronyms.

```
\ifcsundef{gls@#2@entryfmt}%  
{%  
  \defglsentryfmt[#2]{\glsentryfmt}%  
}%  
{}
```

Define sort counter if required:

```
\@gls@defsortcount{#2}%
```

Find out if the final optional argument has been specified, and use it to set the counter associated with this glossary. (Uses `\glscounter` if no optional argument is present.)

```
\@ifnextchar[{\@gls@setcounter{#2}}%  
  {\@gls@setcounter{#2}[\glscounter]}%  
}%  
{%  
  \gls@gobbleopt  
}%  
}
```

`\altnewglossary`

```
\newcommand*{\altnewglossary}[3]{%  
  \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%  
}
```

Only define new glossaries in the preamble:

```
\@onlypreamble{\newglossary}
```

Only define new glossaries before `\makeglossaries`

```
\@onlypremakeg\newglossary
```

`\newglossary` is used to specify the file extensions for the `makeindex` input, output and transcript files. It is written to the auxiliary file by `\newglossary`. Since it is not used by L^AT_EX, `\@newglossary` simply ignores its arguments.

```
\@newglossary
```

```
\newcommand*{\@newglossary}[4]{}
```

Store counter to be used for given glossary type (the first argument is the glossary label, the second argument is the name of the counter):

```
\@gls@setcounter
```

```
\def\@gls@setcounter#1[#2]{%  
  \expandafter\def\csname @glstype@#1@counter\endcsname{#2}%
```

Add counter to xindy list, if not already added:

```
  \ifglsxindy  
    \GlsAddXdyCounters{#2}%  
  \fi  
}
```

Get counter associated with given glossary (the argument is the glossary label):

```
\@gls@getcounter
```

```
\newcommand*{\@gls@getcounter}[1]{%  
  \csname @glstype@#1@counter\endcsname  
}
```

Define the main glossary. This will be the first glossary to be displayed when using `\printglossaries`.

```
\glsdefmain
```

Define the “acronym” glossaries if required.

```
\@gls@do@acronymsdef
```

Define the “symbols”, “numbers” and “index” glossaries if required.

```
\@gls@do@symbolsdef  
\@gls@do@numbersdef  
\@gls@do@indexdef
```

`\newignoredglossary` Creates a new glossary that doesn’t have associated files. This glossary is ignored by any commands that iterate over glossaries, such as `\printglossaries`, and won’t work with commands like `\printglossary`. It’s intended for entries that are so commonly-known they don’t require a glossary.

```
\newcommand*{\newignoredglossary}[1]{%  
  \ifdefempty\@ignored@glossaries  
  {%  
    \protected@edef\@ignored@glossaries{#1}%  
  }%
```

```

{%
  \protected@eappto\@ignored@glossaries{,#1}%
}%
\csgdef{glolist@#1}{,}%
\ifcsundef{gls@#1@entryfmt}%
{%
  \defglsentryfmt[#1]{\glsentryfmt}%
}%
}%
\ifdefempty\@gls@nohyperlist
{%
  \renewcommand*\@gls@nohyperlist{#1}%
}%
{%
  \protected@eappto\@gls@nohyperlist{,#1}%
}%
}

```

`\@ignored@glossaries` List of ignored glossaries.

```
\newcommand*\@ignored@glossaries{}
```

`\ifignoredglossary` Tests if the given glossary is an ignored glossary. Expansion is used in case the first argument is a control sequence.

```

\newcommand*\ifignoredglossary[3]{%
  \protected@edef\@gls@igtype{#1}%
  \expandafter\DTLifinlist\expandafter
  {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
}

```

2.7 Defining new entries

New glossary entries are defined using `\newglossaryentry`. This command requires a label and a key-value list that defines the relevant information for that entry. The definition for these keys follows. Note that the `name`, `description` and `symbol` keys will be sanitized later, depending on the value of the package option `sanitize` (this means that if some of the keys haven't been defined, they can be constructed from the `name` and `description` key before they are sanitized).

name The `name` key indicates the name of the term being defined. This is how the term will appear in the glossary. The `name` key is required when defining a new glossary entry.

```

\define@key{glossentry}{name}{%
  \def\@glo@name{#1}%
}

```

description The `description` key is usually only used in the glossary, but can be made to appear in the text by redefining `\glsentryfmt` or using `\defglsentryfmt`. The

description key is required when defining a new glossary entry. If a long description is required, use `\longnewglossaryentry` instead of `\newglossaryentry`.

```
\define@key{glossentry}{description}{%  
\def\@glo@desc{#1}%  
}
```

`descriptionplural`

```
\define@key{glossentry}{descriptionplural}{%  
\def\@glo@descplural{#1}%  
}
```

`sort` The sort key needs to be sanitized here (the sort key is provided for `makeindex`'s benefit, not for use in the document). The sort key is optional when defining a new glossary entry. If omitted, the value is given by $\langle name \rangle \langle description \rangle$.

```
\define@key{glossentry}{sort}{%  
\def\@glo@sort{#1}}
```

`text` The text key determines how the term should appear when used in the document (i.e. outside of the glossary). If omitted, the value of the `name` key is used instead.

```
\define@key{glossentry}{text}{%  
\def\@glo@text{#1}%  
}
```

`plural` The plural key determines how the plural form of the term should be displayed in the document. If omitted, the plural is constructed by appending `\glspluralsuffix` to the value of the text key.

```
\define@key{glossentry}{plural}{%  
\def\@glo@plural{#1}%  
}
```

`first` The first key determines how the entry should be displayed in the document when it is first used. If omitted, it is taken to be the same as the value of the text key.

```
\define@key{glossentry}{first}{%  
\def\@glo@first{#1}%  
}
```

`firstplural` The `firstplural` key is used to set the plural form for first use, in the event that the plural is required the first time the term is used. If omitted, it is constructed by appending `\glspluralsuffix` to the value of the first key.

```
\define@key{glossentry}{firstplural}{%  
\def\@glo@firstplural{#1}%  
}
```

`\@gls@default@value`

```
\newcommand*{\@gls@default@value}{\relax}
```

symbol The `symbol` key is ignored by most of the predefined glossary styles, and defaults to `\relax` if omitted. It is provided for glossary styles that require an associated symbol, as well as a name and description. To make this value appear in the glossary, you need to redefine `\glossentry`. If you want this value to appear in the text when the term is used by commands like `\gls`, you will need to change `\glentryfmt` (or use for `\defglentryfmt` individual glossaries).

```
\define@key{glossentry}{symbol}{%
\def\@glo@symbol{#1}%
}
```

symbolplural

```
\define@key{glossentry}{symbolplural}{%
\def\@glo@symbolplural{#1}%
}
```

type The `type` key specifies to which glossary this entry belongs. If omitted, the default glossary is used.

```
\define@key{glossentry}{type}{%
\def\@glo@type{#1}}
```

counter The `counter` key specifies the name of the counter associated with this glossary entry:

```
\define@key{glossentry}{counter}{%
\ifcsundef{c@#1}%
{%
\PackageError{glossaries}%
{There is no counter called ‘#1’}%
{%
The counter key should have the name of a valid counter
as its value%
}%
}%
}%
\def\@glo@counter{#1}%
}%
}
```

see The `see` key specifies a list of cross-references

```
\define@key{glossentry}{see}{%
\gls@set@xr@key{see}{\@glo@see}{#1}%
}
```

```
\gls@set@xr@key{<key name>}{<cs>}{<value>}
```

`\gls@set@xr@key`

Assign a cross-reference key. This has a check for an empty value because with `glossaries-extra` and unrestricted document definitions, the `glsdefs` file may contain the cross-referencing fields set to empty.


```

\newcommand*\gls@set@xr@key}[3]{%
  \renewcommand*\gls@xr@key}{#1}%
  \ifstrempy{#3}{\def#2{}}%
  {%
    \gls@checkseeallowed
    \def#2{#3}%
    \@glo@seeautonumberlist
  }%
}

```

`\gls@xr@key`

```

\newcommand*\gls@xr@key}{see}

```

`\gls@checkseeallowed`

```

\newcommand*\gls@checkseeallowed){%
  \@gls@see@noindex
}

```

`\gls@checkseeallowed@preambleonly`

```

\newcommand*\gls@checkseeallowed@preambleonly){%
  \GlossariesWarning
  {'\gls@xr@key' key doesn't have any effect when used in the document
  environment. Move the definition to the preamble
  after \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}

```

`parent` The parent key specifies the parent entry, if required.

```

\define@key{glossentry}{parent}{%
  \def\@glo@parent{#1}}

```

`nonumberlist` The nonumberlist key suppresses or activates the number list for the given entry.

```

\define@choicekey{glossentry}{nonumberlist}{%
  [\gls@nonumberlist@val\gls@nonumberlist@nr]{true,false}[true]%
  {%
    \ifcase\gls@nonumberlist@nr\relax
      \def\@glo@prefix{\glsnonextpages}%
      \@gls@savenonumberlist{true}%
    \else
      \def\@glo@prefix{\glsnextpages}%
      \@gls@savenonumberlist{false}%
    \fi
  }%
}

```

`\gls@noidx@nonumberlist` `\makenoidxglossaries` sets `\@gls@savenonumberlist` to this.

```

\newcommand*\gls@noidx@nonumberlist}[1]{%
  \cslet\@glo@\@glo@label @prenumberlist}\@glo@prefix
}%

```

```

\glsnoidxprenumberlist
    \newcommand{\glsnoidxprenumberlist}[1]{%
        \csuse{glo@glsdetoklabel{#1}@prenumberlist}}

\@gls@savenonumberlist The nonumberlist option isn't saved by default (as it just sets the prefix) which
isn't a problem when the entries are defined in the preamble, but causes a
problem when entries are defined in the document. In this case, the value needs
to be saved so that it can be written to the .glsdefs file.
    \newcommand*{\@gls@savenonumberlist}[1]{

\@gls@initnonumberlist
    \newcommand*{\@gls@initnonumberlist}{}%

\@gls@initnonumberlist
    \newcommand*{\@gls@storenonumberlist}[1]{

\@gls@enablesavenonumberlist Allow the nonumberlist value to be saved.
    \newcommand*{\@gls@enablesavenonumberlist}{%
        \renewcommand*{\@gls@initnonumberlist}{%
            \undef@glo@nonumberlist
        }%
        \renewcommand*{\@gls@savenonumberlist}[1]{%
            \def@glo@nonumberlist{##1}%
        }%
        \renewcommand*{\@gls@storenonumberlist}[1]{%
            \ifdef@glo@nonumberlist
            {%
                \cslet{glo@glsdetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
            }%
            {}%
        }%
        \appto@gls@keymap{,{nonumberlist}{nonumberlist}}%
    }

    Define some generic user keys. (Additional keys can be added by the user.)

user1
    \define@key{glossentry}{user1}{%
        \def@glo@useri{#1}%
    }

user2
    \define@key{glossentry}{user2}{%
        \def@glo@userii{#1}%
    }

user3
    \define@key{glossentry}{user3}{%
        \def@glo@useriii{#1}%
    }

```

```

user4
\define@key{glossentry}{user4}{%
\def\@glo@useriv{#1}%
}

```

```

user5
\define@key{glossentry}{user5}{%
\def\@glo@useriv{#1}%
}

```

```

user6
\define@key{glossentry}{user6}{%
\def\@glo@useriv{#1}%
}

```

`short` This key is provided for use by `\newacronym`. It's not designed for general purpose use, so isn't described in the user manual.

```

\define@key{glossentry}{short}{%
\def\@glo@short{#1}%
}

```

`shortplural` This key is provided for use by `\newacronym`.

```

\define@key{glossentry}{shortplural}{%
\def\@glo@shortpl{#1}%
}

```

`long` This key is provided for use by `\newacronym`.

```

\define@key{glossentry}{long}{%
\def\@glo@long{#1}%
}

```

`longplural` This key is provided for use by `\newacronym`.

```

\define@key{glossentry}{longplural}{%
\def\@glo@longpl{#1}%
}

```

`\@glsnoname` Define command to generate error if name key is missing.

```

\newcommand*\@glsnoname{%
\PackageError{glossaries}{name key required in
\string\newglossaryentry\space for entry '\@glo@label'}{You
haven't specified the entry name}}

```

`\@glsnodesc` Define command to generate error if description key is missing.

```

\newcommand*\@glsnodesc{%
\PackageError{glossaries}
{%
description key required in \string\newglossaryentry\space
for entry '\@glo@label'%
}
}

```

```

    }%
    {%
    You haven't specified the entry description%
    }%
}%

```

`\glsdefaultplural` `\glsdefaultplural` was obsoleted in v3.10a (2013-10-13) and removed in v4.50.

`\gls@missingnumberlist` Define a command to generate warning when numberlist not set.

```

\newcommand*{\gls@missingnumberlist}[1]{%
  ??%
  \ifglssavenumberlist
    \GlossariesWarning{Missing number list for entry '#1'.
      Maybe makeglossaries + rerun required}%
  \else
    \PackageError{glossaries}%
    {Package option 'savenumberlist=true' required}%
    {%
      You must use the 'savenumberlist' package option
      to reference location lists.%
    }%
  \fi
}

```

`\glsdefaultsort` Define command to set default sort.

```

\newcommand*{\glsdefaultsort}{\@glo@name}

```

`\gls@level` Register to increment entry levels.

```

\newcount\gls@level

```

`\@gls@noexpand@field`

```

\newcommand{\@gls@noexpand@field}[3]{%
  \expandafter\global\expandafter
  \let\csname glo@#1@#2\endcsname#3%
}

```

`\@gls@noexpand@fields`

```

\newcommand{\@gls@noexpand@fields}[4]{%
  \ifcsdef{gls@assign@#3@field}
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
    }%
  }%
  {%
    \csuse{gls@assign@#3@field}{#2}{#4}%
  }%
}

```

```

}%
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%

\@gls@noexpand@field{#2}{#3}{\@gls@value}%
}%
{%
\@gls@noexpand@field{#2}{#3}{#4}%
}%
}%
}

```

\@gls@expand@field

```

\newcommand{\@gls@expand@field}[3]{%
\expandafter
\protected@xdef\csname glo@#1@#2\endcsname{#3}%
}

```

@gls@expand@fields

```

\newcommand{\@gls@expand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
{%
\@gls@expand@field{#2}{#3}{#4}%
}%
{%
\csuse{gls@assign@#3@field}{#2}{#4}%
}%
}%
}%
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\@gls@expand@field{#2}{#3}{#1}%
}%
{%
\@gls@expand@field{#2}{#3}{#4}%
}%
}%
}

```

`\@gls@startswithexpandonce`

```
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
  \def\@gls@tmp{#1}%
  \ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
```

```
\gls@assign@field{<def value>}{<label>}{<field>}{<tmp
cs>}
```

`\gls@assign@field`

Assigns an entry field. Expansion performed by default (except for name, symbol and description where backward compatibility required). If `<tmp cs>` is `<@gls@default@value>`, `<def value>` is used instead.

```
\let\gls@assign@field\@gls@expand@fields
```

`\glsexpandfields` Fully expand values when assigning fields (except for specific fields that are overridden by `\glssetnoexpandfield`).

```
\newcommand*{\glsexpandfields}{%
  \let\gls@assign@field\@gls@expand@fields
}
```

`\glsnoexpandfields` Don't expand values when assigning fields (except for specific fields that are overridden by `\glssetexpandfield`).

```
\newcommand*{\glsnoexpandfields}{%
  \let\gls@assign@field\@gls@noexpand@fields
}
```

`\newglossaryentry` Define `\newglossaryentry {<label>}{<key-val list>}`. There are two required fields in `<key-val list>`: name (or parent) and description. (See above.)

```
\newrobustcmd{\newglossaryentry}[2]{%
```

Check to see if this glossary entry has already been defined:

```
\glsdoifnoexists{#1}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
```

`\gls@defdocnewglossaryentry` The definition of `\newglossaryentry` is changed at the start of the document environment. As from v4.47 the see key now works in the document environment.

```
\newcommand*{\gls@defdocnewglossaryentry}{%
  \let\newglossaryentry\new@glossaryentry
}
```

`\provideglossaryentry` Like `\newglossaryentry` but does nothing if the entry has already been defined.

```

\newrobustcmd{\provideglossaryentry}[2]{%
  \ifglentryexists{#1}%
  {}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\@onlypreamble{\provideglossaryentry}

```

`\gls@glsdefs@labellist` List of labels of entries defined in `.glsdefs` file.

```
\newcommand*{\gls@glsdefs@labellist}{}
```

`\gls@glsdefs@addlabel` Save label of entry defined in `.glsdefs` file.

```
\newcommand*{\gls@glsdefs@addlabel}[1]{\listxadd{\gls@glsdefs@labellist}{#1}}
```

`\gls@docdef@labellist` List of labels of entries defined in the document environment.

```
\newcommand*{\gls@docdef@labellist}{}
```

`\gls@docdef@addlabel` Save label of entry defined in the document environment.

```
\newcommand*{\gls@docdef@addlabel}[1]{\listxadd{\gls@docdef@labellist}{#1}}
```

`\new@glossaryentry` For use in document environment. This opens the `.glsdefs` file, if not already open, so that the entry definition can be saved for the next \LaTeX run. This means that any glossaries at the start of the document can access the entry information.

```

\newrobustcmd{\new@glossaryentry}[2]{%
  \ifundef\@gls@deffile
  {%
    \global\newwrite\@gls@deffile
    \immediate\openout\@gls@deffile=\jobname.glsdefs
  }%
  {}%
}

```

Don't check for existence since if the entry has been defined in the `.glsdefs` it will need to be redefined in case the entry has been updated, but keep track of which entries have been defined in the document environment to prevent duplicate definitions.

```

\ifglentryexists{#1}%
{%
  \xifinlist{#1}{\gls@docdef@labellist}%
  {%

```

This entry has already been defined in the document environment.

```

  \GlossariesWarning{Entry ‘#1’ has already been defined in the
    document environment (ignoring new definition)}%
  }%
  {}%
  \xifinlist{#1}{\gls@glsdefs@labellist}%
  {%

```

This entry was defined in the `.glsdefs` file so update the definition in case it has been changed.

```
\gls@defglossaryentry{#1}{#2}%
\gls@docdef@addlabel{#1}%
\@gls@writedef{#1}%
}%
{%
```

This entry wasn't defined in the `.glsdefs` file so it must have been defined in the preamble.

```
\GlossariesWarning{Entry '#1' has already been defined in
the preamble (in which case remove the duplicate definition)
or you have just updated to glossaries.sty
v4.47 (in which case delete '\jobname.glsdefs')}%
}%
}%
}%
{%
```

This entry hasn't been defined (new entry since last build).

```
\gls@defglossaryentry{#1}{#2}%
\gls@docdef@addlabel{#1}%
\@gls@writedef{#1}%
}%
}
```

At the start of the document input the `.glsdefs` file if it exists. This is now done by `\gls@begindocdefs`, which is redefined by `glossaries-extra`, so that this step can be skipped to avoid loading an obsolete `.glsdefs` file if the user switches to `glossaries-extra` with `docdef=restricted`.

```
\AtBeginDocument{\gls@begindocdefs}
```

The end of the document needs to check if the `.glsdefs` file has been opened, in which case it needs to be closed.

```
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
```

`\gls@begindocdefs` Input the `.glsdefs` file if it exists and enable document definitions if permitted.

```
\newcommand*{\gls@begindocdefs}{%
\@gls@enablesavenonumberlist
\edef\@gls@restoreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
\makeatletter
\InputIfFileExists{\jobname.glsdefs}{\@gls@restoreat}{\@gls@restoreat}
\undef\@gls@restoreat
\gls@defdocnewglossaryentry
}
```

`\glsdefs@newdocentry` Used to define entries in the `.glsdefs` file.

```
\newcommand*{\glsdefs@newdocentry}[2]{%
\ifglsentryexists{#1}%
```



```

{\GlossariesWarning{entry ‘#1’ has already been defined in the preamble
  (if you have moved the definition there from the document
  environment, delete \jobname.glsdefs)}}%
{%
  \gls@glsdefs@addlabel{#1}%
  \gls@defglossaryentry{#1}{#2}%
}%
}

```

`\gls@writedef` Writes glossary entry definition to `\gls@deffile`. Allow for the possibility that the entry definition may have been moved from the document environment to the preamble, so warn rather than generate an error.

```

\newcommand*{\gls@writedef}[1]{%
  \immediate\write\gls@deffile
  {%
    \string\glsdefs@newdocentry{\glsdetoklabel{#1}}\glspercentchar^^J%
    \expandafter@gobble\string\{\glspercentchar%
  }%
}

```

Write key value information:

```

\@for\gls@map:=\gls@keymap\do
{%
  \letcs\glo@value{glo@\glsdetoklabel{#1}}\expandafter\@secondoftwo\gls@map}%
  \ifdef\glo@value
  {%
    \@onelevel@sanitize\glo@value
    \immediate\write\gls@deffile
    {%
      \expandafter\@firstoftwo\gls@map
      =\expandafter@gobble\string\{\glo@value\expandafter@gobble\string\},%
      \glspercentchar
    }%
  }%
  {}%
}%
}

```

Provide hook:

```

\gls@writedefhook
\immediate\write\gls@deffile
{%
  \glspercentchar^^J%
  \expandafter@gobble\string\}\glspercentchar^^J%
}%
}

```

`\gls@keymap` List of entry definition key names and corresponding tag in control sequence used to store the value.

```

\newcommand*{\gls@keymap}{%
  {name}{name},%
  {sort}{sortvalue},% unescaped sort value
}

```

```

{type}{type},%
{first}{first},%
{firstplural}{firstpl},%
{text}{text},%
{plural}{plural},%
{description}{desc},%
{descriptionplural}{descplural},%
{symbol}{symbol},%
{symbolplural}{symbolplural},%
{user1}{useri},%
{user2}{userii},%
{user3}{useriii},%
{user4}{useriv},%
{user5}{userv},%
{user6}{uservi},%
{long}{long},%
{longplural}{longpl},%
{short}{short},%
{shortplural}{shortpl},%
{counter}{counter},%
{parent}{parent}%
}

```

`\@gls@fetchfield`

```
\@gls@fetchfield{<cs>}{<field>}
```

Fetches the internal field label from the given user *<field>* and stores in *<cs>*.

```
\newcommand*{\@gls@fetchfield}[2]{%
```

Ensure user field name is fully expanded

```
\edef\@gls@thisval{#2}%
```

Iterate through known mappings until we find the one for this field.

```
\@for\@gls@map:=\@gls@keymap\do{%
\edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
\ifdefequal{\@this@key}{\@gls@thisval}%
{%
```

Found it.

```
\edef#1{\expandafter\@secondoftwo\@gls@map}%
```

Break out of loop.

```
\@endfortrue
}%
}%
}%
}
```

`\glsaddstoragekey`

```
\glsaddstoragekey{<key>}{<default value>}{<no link cs>}
```

Similar to `\glsaddkey` but intended for keys whose values aren't explicitly used in the document, but might be required behind the scenes by other commands.

```
\newcommand*\glsaddstoragekey{\@ifstar\sglsaddstoragekey\glsaddstoragekey}
```

Starred version switches on expansion for this key.

```
\newcommand*\@sglsaddstoragekey[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
  {}%
  \@glsaddstoragekey{#1}%
}
```

Unstarred version doesn't override default expansion.

```
\newcommand*\@glsaddstoragekey[3]{%
```

Check the specified key doesn't already exist.

```
\key@ifundefined{glossentry}{#1}%
{%
```

Set up the key.

```
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
```

Set the default value.

```
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
```

Assignment code.

```
\appto\@newglossaryentryposthook{%
  \letcs{\@glo@tmp}{@glo@#1}%
  \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
```

Define the no-link commands.

```
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
}%
{%
  \PackageError{glossaries}{Key '#1' already exists}{}%
}%
}
```

```
\glsaddkey{<key>}{<default value>}{<no link cs>}{<no link
ucfirst
cs>}{<link cs>}{<link ucfirst cs>}{<link allcaps cs>}
```

`\glsaddkey`

Allow user to add their own custom keys.

```
\newcommand*\glsaddkey{\@ifstar\sglsaddkey\glsaddkey}
```

Starred version switches on expansion for this key.

```
\newcommand*\@sglsaddkey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
  {}%
  \@glsaddkey{#1}%
}
```

Unstarred version doesn't override default expansion.

```
\newcommand*\@glsaddkey}[7]{%
```

Check the specified key doesn't already exist.

```
\key@ifundefined{glossentry}{#1}%
{%
```

Set up the key.

```
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
```

Set the default value.

```
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
```

Assignment code.

```
\appto\@newglossaryentryposthook{%
  \letcs{@glo@tmp}{@glo@#1}%
  \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
```

Define the no-link commands.

```
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
\newcommand*{#4}[1]{\@Glsentryfield{##1}{#1}}%
```

Now for the commands with links. First the version with no case change:

```
\ifcsdef{@gls@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#5' as helper command
  '\expandafter\string\csname @gls@user@#1\endcsname' already exists}%
  {}%
}%
{%
  \expandafter\newcommand\expandafter*\expandafter
  {\csname @gls@user@#1\endcsname}[2][1]{%
    \new@ifnextchar[%
      {\csuse{@gls@user@#1@}{##1}{##2}}%
      {\csuse{@gls@user@#1@}{##1}{##2}[1]}%
    \csdef{@gls@user@#1@}##1##2[##3]{%

```

```

    \@gls@field@link{##1}{##2}{#3{##2}##3}%
  }%
  \newrobustcmd*{#5}{%
    \expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
  }%

```

Next the version with the first letter converted to upper case:

```

\ifcsdef{@Gls@user@#1@}%
{
  \PackageError{glossaries}%
  {Can't define '\string#6' as helper command
  '\expandafter\string\csname @Gls@user@#1\endcsname' already exists}%
  {}%
}%
{
  \expandafter\newcommand\expandafter*\expandafter
  {\csname @Gls@user@#1\endcsname}[2][]{%
    \new@ifnextchar[%
      {\csuse{@Gls@user@#1@}{##1}{##2}}%
      {\csuse{@Gls@user@#1@}{##1}{##2}[ ]}}%
  \csdef{@Gls@user@#1@}##1##2[##3]{%
    \@gls@field@link{##1}{##2}{#4{##2}##3}%
  }%
  \newrobustcmd*{#6}{%
    \expandafter\@gls@hyp@opt\csname @Gls@user@#1\endcsname}%
  }%

```

Finally the all caps version:

```

\ifcsdef{@GLS@user@#1@}%
{
  \PackageError{glossaries}%
  {Can't define '\string#7' as helper command
  '\expandafter\string\csname @GLS@user@#1\endcsname' already exists}%
  {}%
}%
{
  \expandafter\newcommand\expandafter*\expandafter
  {\csname @GLS@user@#1\endcsname}[2][]{%
    \new@ifnextchar[%
      {\csuse{@GLS@user@#1@}{##1}{##2}}%
      {\csuse{@GLS@user@#1@}{##1}{##2}[ ]}}%
  \csdef{@GLS@user@#1@}##1##2[##3]{%
    \@gls@field@link{##1}{##2}{\glsuppercase{#3{##2}##3}}%
  }%
  \newrobustcmd*{#7}{%
    \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
  }%

```

Add mappings.

```

\glsmfuaddmap{#3}{#4}%
\glsmfuaddmap{#5}{#6}%
\glsmfublocker{#7}%
}%
{%
\PackageError{glossaries}{Key ‘#1’ already exists}{}%
}%
}

```

`\glsfieldxdef`

```
\glsfieldxdef{<label>}{<field>}{<definition>}
```

```

\newrobustcmd{\glsfieldxdef}[3]{%
\glsdoifexists{#1}%
{%
\protected@edef\glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\glo@label @#2}%
{%
\protected@csxdef{glo@\glo@label @#2}{#3}%
}%
{%
\PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
}%
}%
}

```

`\glsfielddedef`

```
\glsfielddedef{<label>}{<field>}{<definition>}
```

```

\newrobustcmd{\glsfielddedef}[3]{%
\glsdoifexists{#1}%
{%
\protected@edef\glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\glo@label @#2}%
{%
\protected@csedef{glo@\glo@label @#2}{#3}%
}%
{%
\PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
}%
}%
}

```

`\glsfieldgdef`

```
\glsfieldgdef{<label>}{<field>}{<definition>}
```

```

\newrobustcmd{\glsfieldgdef}[3]{%
\glsdoifexists{#1}%
{%
\protected@edef\@glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\@glo@label @#2}%
{%
\expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
}%
}%
\PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
}%
}

```

\glsfielddef

```
\glsfielddef{<label>}{<field>}{<definition>}
```

```

\newrobustcmd{\glsfielddef}[3]{%
\glsdoifexists{#1}%
{%
\protected@edef\@glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\@glo@label @#2}%
{%
\expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
}%
}%
\PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
}%
}

```

\glsfieldfetch

```
\glsfieldfetch{<label>}{<field>}{<cs>}
```

Fetches the value of the given field and stores in the given control sequence.

```

\newrobustcmd{\glsfieldfetch}[3]{%
\glsdoifexists{#1}%
{%
\protected@edef\@glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\@glo@label @#2}%
{%
\letcs#3{glo@\@glo@label @#2}%
}%
}%
\PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
}%
}

```

```
}%
}
```

```
\glsunexpandedfieldvalue{<label>}{<field>}
```

`\glsunexpandedfieldvalue`

For use in expandable contexts where the field value is required, but the contents should not be expanded. The `<field>` argument must be the internal field name.

```
\newrobustcmd*{\glsunexpandedfieldvalue}[2]{%
  \ifcsname glo@\glsdetoklabel{#1}@#2\endcsname
  \expandafter\expandonce\csname glo@\glsdetoklabel{#1}@#2\expandafter\endcsname
  \fi
}
```

```
\ifglsfieldef{<label>}{<field>}{<string>}{<true>}{<false>}
```

`\ifglsfieldef`

Tests if the value of the given field is equal to the given string.

```
\newrobustcmd{\ifglsfieldef}[5]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\glo@label @#2}%
    {%
      \ifcsstring{glo@\glo@label @#2}{#3}{#4}{#5}%
    }%
    {%
      \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
    }%
  }%
}
```

```
\ifglsfielddefeq{<label>}{<field>}{<command>}{<true>}
{<false>}
```

`\ifglsfielddefeq`

Tests if the value of the given field is equal to the replacement text of the given command.

```
\newrobustcmd{\ifglsfielddefeq}[5]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\glo@label @#2}%
    {%
```

Use `\ifdefstrequal` not `\ifcsstrequal` as the #3 argument should be a command.

```
\expandafter\ifdefstrequal\csname glo@\glo@label @#2\endcsname{#3}{#4}{#5}%
```



```

    }%
    {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
    }%
  }%
}

```

```

\ifglsfieldcseq{<label>}{<field>}{<cs name>}{<true>}
{<false>}

```

\ifglsfieldcseq

As above but uses \ifcsstrequal instead of \ifdefstrequal

```

\newcommand{\ifglsfieldcseq}[5]{%
\glsdoifexists{#1}%
{%
\protected@edef\glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\glo@label @#2}%
{%
\ifcsstrequal{glo@\glo@label @#2}{#3}{#4}{#5}%
}%
}%
\PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
}%
}

```

\gls.writedefhook

```

\newcommand*{\gls.writedefhook}{}

```

\gls@assign@desc

```

\newcommand*{\gls@assign@desc}[1]{%
\gls@assign@field{#1}{desc}{\@glo@desc}%
\gls@assign@field{\@glo@desc}{#1}{descplural}{\@glo@descplural}%
}

```

\longnewglossaryentry

```

\newcommand{\longnewglossaryentry}[3]{%
\glsdoifnoexists{#1}%
{%
\bgroup
\let\@org@newglossaryentryprehook\@newglossaryentryprehook
\long\def\@newglossaryentryprehook{%
\long\def\@glo@desc{#3\leavevmode\unskip\nopostdesc}%
\@org@newglossaryentryprehook
}%
\renewcommand*{\gls@assign@desc}[1]{%
\global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
\global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@desc}%
}
}

```

```

        \gls@defglossaryentry{#1}{#2}%
    \egroup
}
}

```

Only allowed in the preamble. (Otherwise a long description could cause problems when writing the entry definition to the temporary file.)

```
\@onlypreamble{\longnewglossaryentry}
```

`\longprovideglossaryentry` As the above but only defines the entry if it doesn't already exist.

```

\newcommand{\longprovideglossaryentry}[3]{%
  \ifglentryexists{#1}{}%
  {\longnewglossaryentry{#1}{#2}{#3}}%
}
\@onlypreamble{\longprovideglossaryentry}

```

```
\gls@defglossaryentry{<label>}{<key-val list>}
```

`\gls@defglossaryentry`

Defines a new entry without checking if it already exists.

```
\newcommand{\gls@defglossaryentry}[2]{%
```

Prevent any further use of `\GlsSetQuote`:

```
\let\GlsSetQuote\gls@nosetquote
```

Store label

```
\protected@edef\glo@label{\glsdetoklabel{#1}}%
```

Provide a means for user defined keys to reference the label:

```
\let\glslabel\glo@label
```

Set up defaults. If the name or description keys are omitted, an error will be generated.

```
\let\glo@name\gls@name
```

```
\let\glo@desc\gls@desc
```

```
\let\glo@descplural\gls@default@value
```

```
\let\glo@type\gls@default@value
```

```
\let\glo@symbol\gls@default@value
```

```
\let\glo@symbolplural\gls@default@value
```

```
\let\glo@text\gls@default@value
```

```
\let\glo@plural\gls@default@value
```

Using `\let` instead of `\def` to make later comparison avoid expansion issues. (Thanks to Ulrich Diez for suggesting this.)

```
\let\glo@first\gls@default@value
```

```
\let\glo@firstplural\gls@default@value
```

Set the default sort:

```
\let\@glo@sort\@gls@default@value
```

Set the default counter:

```
\let\@glo@counter\@gls@default@value
```

```
\def\@glo@see{ }%
```

```
\def\@glo@parent{ }%
```

```
\def\@glo@prefix{ }%
```

Initialise nonnumberlist setting if we're in the document environment.

```
\@gls@initnonnumberlist
```

```
\def\@glo@useri{ }%
```

```
\def\@glo@userii{ }%
```

```
\def\@glo@useriii{ }%
```

```
\def\@glo@useriv{ }%
```

```
\def\@glo@userv{ }%
```

```
\def\@glo@uservi{ }%
```

```
\def\@glo@short{ }%
```

```
\def\@glo@shortpl{ }%
```

```
\def\@glo@long{ }%
```

```
\def\@glo@longpl{ }%
```

Add start hook in case another package wants to add extra keys.

```
\@newglossaryentryprehook
```

Extract key-val information from third parameter:

```
\setkeys{glossentry}{#2}%
```

Check there is a default glossary.

```
\ifundef\glsdefaulttype
```

```
{%
```

```
\PackageError{glossaries}%
```

```
{No default glossary type (have you used 'nomain' by mistake?)}%
```

```
{If you use package option 'nomain' you must define
```

```
a new glossary before you can define entries}%
```

```
}%
```

```
{ }%
```

Assign type. This must be fully expandable

```
\gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
```

```
\protected@edef\@glo@type{\glsentrytype{\@glo@label}}%
```

Check to see if this glossary type has been defined, if it has, add this label to the relevant list, otherwise generate an error.

```
\ifcsundef{glolist@\@glo@type}%
```

```
{%
```

```
\PackageError{glossaries}%
```

```
{Glossary type '\@glo@type' has not been defined}%
```

```

        {You need to define a new glossary type, before making entries
        in it}%
    }%
    {%
Check if it's an ignored glossary
    \ifignoredglossary\@glo@type
    {%
The description may be omitted for an entry in an ignored glossary.
    \ifx\@glo@desc\@glsnodesc
    \let\@glo@desc\@empty
    \fi
    }%
    {%
    }%
    \protected@edef\@glo@list@\csname glist@\@glo@type\endcsname}%

    \expandafter\protected@xdef\csname glist@\@glo@type\endcsname{%
    \@glo@list@\@glo@label},}%
    }%

Initialise level to 0.
    \gls@level=0\relax
Has this entry been assigned a parent?
    \ifx\@glo@parent\@empty
Doesn't have a parent. Set \glo@<label>@parent to empty.
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
    \else
Has a parent. Check to ensure this entry isn't its own parent.
    \ifdefequal\@glo@label\@glo@parent%
    {%
        \PackageError{glossaries}{Entry '@@glo@label' can't be its own parent}{}%
        \def\@glo@parent{}%
        \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
    }%
    {%
Check the parent exists:
    \ifglsentryexists{\@glo@parent}%
    {%
Parent exists. Set \glo@<label>@parent.
    \expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
    \@glo@parent}%

Determine level.
    \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
    \advance\gls@level by 1\relax

```

If name hasn't been specified, use same as the parent name

```
\ifx\@glo@name\@glsnoname
\expandafter\let\expandafter\@glo@name
\csname glo@\@glo@parent @name\endcsname
```

If name and plural haven't been specified, use same as the parent

```
\ifx\@glo@plural\@gls@default@value
\expandafter\let\expandafter\@glo@plural
\csname glo@\@glo@parent @plural\endcsname
\fi
\fi
}%
{%
```

Parent doesn't exist, so issue an error message and change this entry to have no parent

```
\PackageError{glossaries}%
{%
  Invalid parent '@glo@parent'
  for entry '@glo@label' - parent doesn't exist%
}%
{%
  Parent entries must be defined before their children%
}%
\def\@glo@parent{}%
\expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
}%
}%
\fi
```

Set the level for this entry

```
\expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%
```

Define commands associated with this entry:

```
\gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
\letcs\@glo@sort{glo@\@glo@label @sortvalue}%
\gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @text\endcsname\glspluralsuffix}%
  {\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @text\endcsname}%
  {\@glo@label}{first}{\@glo@first}%
```

If first has been specified, make the default by appending `\glspluralsuffix`, otherwise make the default the value of the plural key.

```
\ifx\@glo@first\@gls@default@value
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @plural\endcsname}%
  {\@glo@label}{firstpl}{\@glo@firstplural}%
\else
```

```

\expandafter\gls@assign@field\expandafter
  {\csname glo@\glo@label @first\endcsname\glspluralsuffix}%
  {\@glo@label}{firstpl}{\@glo@firstplural}%
\fi

\ifcsundef{@glo@type@\glo@type @counter}%
{%
  \def\glo@defaultcounter{\glscounter}%
}%
{%
  \letcs\@glo@defaultcounter{@glo@type@\glo@type @counter}%
}%
\gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
\gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
\gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
\gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
\gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
\gls@assign@field{}{\@glo@label}{userv}{\@glo@userv}%
\gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
\gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
\gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
\gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
\gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%
\ifx\@glo@name\@glsnoname
  \@glsnoname
  \let\@glo@name\@gls@default@value
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%

```

Set default numberlist if not defined:

```

\ifcsundef{glo@\glo@label @numberlist}%
{%
  \csxdef{glo@\glo@label @numberlist}{%
    \noexpand\@gls@missingnumberlist{\@glo@label}}%
}%
{%
  }%

```

Store nonumberlist setting if we're in the document environment.

```

\@gls@storenonumberlist{\@glo@label}%

```

The smaller and smallcaps options set the description to \@glo@first. Need to check for this, otherwise it won't get expanded if the description gets sanitized.

```

\def\@glo@@desc{\@glo@first}%
\ifx\@glo@desc\@glo@@desc
  \let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@glsnodesc
  \@glsnodesc
  \let\@glo@desc\@gls@default@value
\fi

```

```
\gls@assign@desc{\@glo@label}%
```

Set the sort key for this entry:

```
\gls@defsort{\@glo@type}{\@glo@label}%
```

```
\def\@glo@symbol{\@glo@text}%
```

```
\ifx\@glo@symbol\@glo@symbol
```

```
\let\@glo@symbol\@glo@text
```

```
\fi
```

```
\gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
```

```
\expandafter
```

```
\gls@assign@field\expandafter
```

```
{\csname glo@\@glo@label @symbol\endcsname}
```

```
{\@glo@label}{symbolplural}{\@glo@symbolplural}%
```

Define an associated boolean variable to determine whether this entry has been used yet (needs to be defined globally):

```
\expandafter\protected@xdef\csname glo@\@glo@label @flagfalse\endcsname{%
```

```
\noexpand\global
```

```
\noexpand\let\expandafter\noexpand
```

```
\csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
```

```
}%
```

```
\expandafter\protected@xdef\csname glo@\@glo@label @flagtrue\endcsname{%
```

```
\noexpand\global
```

```
\noexpand\let\expandafter\noexpand
```

```
\csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
```

```
}%
```

```
\csname glo@\@glo@label @flagfalse\endcsname
```

Sort out any cross-referencing if required.

```
\@glo@autosee
```

Determine and store main part of the entry's index format.

```
\ifignoredglossary\@glo@type
```

```
{%
```

```
\csdef{glo@\@glo@label @index}{}
```

```
}
```

```
{%
```

```
\do@glo@storeentry{\@glo@label}%
```

```
}%
```

Define entry counters if enabled:

```
\@newglossaryentry@defcounters
```

Add end hook in case another package wants to add extra keys.

```
\@newglossaryentryposthook
```

```
}
```

`\@glo@autosee` Automatically implement `\glssee`.

```
\newcommand*{\@glo@autosee}{%
```

```
\ifdefvoid\@glo@see{}
```

```
{%
```

```

        \protected@edef\do@glsee{%
        \noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
        \noexpand\expandafter\noexpand\@glsee\noexpand\@glo@list{\@glo@label}}%
        \do@glsee
    }%
    \@glo@autoseehook
}%

\@glo@autoseehook

\newcommand*{\@glo@autoseehook}{-}

\@newglossaryentryprehook Allow extra information to be added to glossary entries:
\newcommand*{\@newglossaryentryprehook}{-}

\@newglossaryentryposthook Allow extra information to be added to glossary entries:
\newcommand*{\@newglossaryentryposthook}{-}

newglossaryentry@defcounters

\newcommand*{\@newglossaryentry@defcounters}{-}

\glsmoveentry Moves entry whose label is given by first argument to the glossary named in the
second argument.
\newcommand*{\glsmoveentry}[2]{%
\protected@edef\@glo@thislabel{\glsdetoklabel{#1}}%
\protected@edef\@glo@type{\csname glo@\@glo@thislabel @type\endcsname}%
\def\@glo@list{,%}
\for@glseentries[\@glo@type]{\@glo@label}%
{
\ifdefequal\@glo@thislabel\@glo@label
}{\protected@eappto\@glo@list{\@glo@label,}}%
}%
\cslet\@glo@list[\@glo@type]{\@glo@list}%
\csdef{glo@\@glo@thislabel @type}{#2}%
}

\@glossaryentryfield Indicate what command should be used to display each entry in the glossary.
(This enables the glossaries-accsupp package to use \accsuppglossaryentryfield
instead.)
\ifglxindy
\newcommand*{\@glossaryentryfield}{\string\glossentry}
\else
\newcommand*{\@glossaryentryfield}{\string\glossentry}
\fi

\@glossarysubentryfield Indicate what command should be used to display each subentry in the glossary.
(This enables the glossaries-accsupp package to use \accsuppglossarysubentryfield
instead.)
\ifglxindy

```



```

\newcommand*{\@glossarysubentryfield}{%
  \string\subglossentry}
\else
\newcommand*{\@glossarysubentryfield}{%
  \string\subglossentry}
\fi

```

```
\@glo@storeentry{<label>}
```

\@glo@storeentry

Determine the format to write the entry in the glossary output (.glo) file. The argument is the entry's label (should already have been de-tok'ed if required). The result is stored in \glo@<label>@index, where <label> is the entry's label. (This doesn't include any formatting or location information.)

```
\newcommand{\@glo@storeentry}[1]{%
```

Escape makeindex/xindy special characters in the label:

```

\protected@edef\@glo@esclabel{#1}%
\@gls@checkmkidxchars\@glo@esclabel

```

Get the sort string and escape any special characters

```

\protected@edef\@glo@sort{\csname glo@#1@sort\endcsname}%
\@gls@checkmkidxchars\@glo@sort

```

Same again for the name string. Escape any special characters in the prefix

```
\@gls@checkmkidxchars\@glo@prefix
```

Get the parent, if one exists

```
\protected@edef\@glo@parent{\csname glo@#1@parent\endcsname}%
```

Write the information to the glossary file.

```
\ifglxindy
```

Store using xindy syntax.

```
\ifx\@glo@parent\@empty
```

Entry doesn't have a parent

```

\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
  (\string\@glo@sort\string" %
  \string\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %
}%
\else

```

Entry has a parent

```

\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
  \csname glo@\@glo@parent @index\endcsname
  (\string\@glo@sort\string" %
  \string\@glo@prefix\@glossarysubentryfield
  {\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
}%

```

```

\fi
\else

```

Store using makeindex syntax.

```
\ifx\@glo@parent\@empty
```

Sanitize \@glo@prefix

```
\@onelevel@sanitize\@glo@prefix
```

Entry doesn't have a parent

```
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%  
  \@glo@sort\@gls@actualchar\@glo@prefix  
  \@glossaryentryfield{\@glo@esclabel}%  
}%
```

```
\else
```

Entry has a parent

```
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%  
  \csname glo@\@glo@parent @index\endcsname\@gls@levelchar  
  \@glo@sort\@gls@actualchar\@glo@prefix  
  \@glossarysubentryfield  
  {\csname glo@#1@level\endcsname}{\@glo@esclabel}%  
}%
```

```
\fi
```

```
\fi
```

```
}
```

2.8 Resetting and unsetting entry flags

Each glossary entry is assigned a conditional of the form `\ifglo@<label>@flag` which determines whether or not the entry has been used (see also `\ifglsused` defined below). These flags can be set and unset using the following macros, but first we need to know if we're in `amsmath`'s `align` environment's measuring pass.

```
\@gls@if@measuring
```

```
\newcommand*{\@gls@if@measuring}[2]{%  
  \ifmeasuring@  
    #1%  
  \else  
    #2%  
  \fi  
}
```

```
\gls@org@refstepcounter
```

```
\let\gls@org@refstepcounter\refstepcounter  
\let\gls@org@label\label
```

```
\gls@if@measuring
```

```
\newcommand{\gls@if@measuring}[2]{#2}
```

If amsmath has been loaded, set `\gls@if@measuring` to `\@gls@if@measuring`

```
\AtBeginDocument{%
  \ifpackageloaded{amsmath}%
  {\let\gls@if@measuring\@gls@if@measuring}%
  {}%
  \let\gls@org@refstepcounter\refstepcounter
  \let\gls@org@label\label
}
```

`\gls@ifnotmeasuring`

```
\newcommand*\gls@ifnotmeasuring[1]{\glsifmeasuring{#1}}
```

`\glsifmeasuring`

```
\newcommand*\glsifmeasuring{\gls@if@measuring}
```

`\gls@local@stepcounter`

```
\newcommand*\gls@local@stepcounter[1]{%
  \ifcsdef{c#1}%
  {%
    \epreto\gls@measuring@reset{\noexpand\setcounter{#1}{\number\value{#1}}}%
    \expandafter\advance\csname c#1\endcsname by \@ne
  }%
  {}%
}
```

`\gls@start@measuring` It's best to scope this, but still provide some support to switch on and off.

```
\newcommand{\gls@start@measuring}{%
  \ifx\glsifmeasuring\@firstoftwo
  \else
    \let\gls@org@target\glstarget
    \let\gls@org@link\@glslink
    \def\gls@measuring@reset{}%
  \fi
  \let\glsifmeasuring\@firstoftwo
  \let\refstepcounter\gls@local@stepcounter
  \let\label\@gobble
  \let\glstarget\@secondoftwo
  \let\@glslink\@secondoftwo
}
```

If datatool-base v3.0 available, add to measuring hook:

```
\ExplSyntaxOn
\tl_if_exist:NT \l_datatool_measure_hook_tl
{
  \tl_put_right:Nn \l_datatool_measure_hook_tl
  { \gls@start@measuring }
}
\ExplSyntaxOff
```

`\gls@stop@measuring`

```
\newcommand{\gls@stop@measuring}{%
  \renewcommand*{\gls@ifmeasuring}{\gls@if@measuring}%
  \let\refstepcounter\gls@org@refstepcounter
  \let\label\gls@org@label
  \let\glstarget\gls@org@target
  \let\@glslink\gls@org@link
  \gls@measuring@reset
  \def\gls@measuring@reset{%
}
```

`\glsmeasureheight`

```
\newcommand{\glsmeasureheight}[2]{%
  \settoheight{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
```

`\glsmeasuredepth`

```
\newcommand{\glsmeasuredepth}[2]{%
  \settodepth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
```

`\glsmeasurewidth`

```
\newcommand{\glsmeasurewidth}[2]{%
  \settowidth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
```

`\glspatchtabularx` Patch `\TX@trial` (as per David Carlisle's answer in <http://tex.stackexchange.com/a/94895>). This does nothing if `\TX@trial` hasn't been defined.

```
\def\@gls@patchtabularx#1\hbox#2#3!!{%
  \def\TX@trial##1{##1\hbox{\gls@start@measuring#2\gls@stop@measuring}##3}%
}
\newcommand*\glspatchtabularx{%
  \ifdef\TX@trial
  {%
    \expandafter\@gls@patchtabularx\TX@trial{##1}!!%
    \let\glspatchtabularx\relax
  }%
  {}%
}
```

`\glsreset` The command `\glsreset{<label>}` can be used to set the entry flag to indicate that it hasn't been used yet. The required argument is the entry label.

```
\newcommand*\glsreset[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glsreset{#1}%
    }%
  }
```

```
    }%
  }
```

`\glslocalreset` As above, but with only a local effect:

```
\newcommand*\glslocalreset[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glslocalreset{#1}%
    }%
  }%
}
```

`\glsunset` The command `\glsunset{<label>}` can be used to set the entry flag to indicate that it has been used. The required argument is the entry label.

```
\newcommand*\glsunset[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glsunset{#1}%
    }%
  }%
}
```

`\glslocalunset` As above, but with only a local effect:

```
\newcommand*\glslocalunset[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glslocalunset{#1}%
    }%
  }%
}
```

`\@glslocalunset` Local unset. This defaults to just `\@@glslocalunset` but is changed by `\glsenableentrycount`.

```
\newcommand*\@glslocalunset{\@@glslocalunset}
```

`\@@glslocalunset` Local unset without checks.

```
\newcommand*\@@glslocalunset[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
}
```

`\@glsunset` Global unset. This defaults to just `\@@glsunset` but is changed by `\glsenableentrycount`.

```
\newcommand*\@glsunset{\@@glsunset}
```

`\@@glsunset` Global unset without checks.

```
\newcommand*{\@@glsunset}[1]{%
  \expandafter\global\csname glo@\glsdetoklabel{#1}@flagtrue\endcsname
}
```

`\@glslocalreset` Local reset. This defaults to just `\@@glslocalreset` but is changed by `\glsenableentrycount`.

```
\newcommand*{\@glslocalreset}{\@@glslocalreset}
```

`\@@glslocalreset` Local reset without checks.

```
\newcommand*{\@@glslocalreset}[1]{%
  \expandafter\let\csname ifglo@\glsdetoklabel{#1}@flag\endcsname\iffalse
}
```

`\@glsreset` Global reset. This defaults to just `\@@glsreset` but is changed by `\glsenableentrycount`.

```
\newcommand*{\@glsreset}{\@@glsreset}
```

`\@@glsreset` Global reset without checks.

```
\newcommand*{\@@glsreset}[1]{%
  \expandafter\global\csname glo@\glsdetoklabel{#1}@flagfalse\endcsname
}
```

Reset all entries for the named glossaries (supplied in a comma-separated list). Syntax: `\glsresetall[<glossary-list>]`

`\glsresetall`

```
\newcommand*{\glsresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsreset{\@glsentry}%
  }%
}
```

As above, but with only a local effect:

`\glslocalresetall`

```
\newcommand*{\glslocalresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalreset{\@glsentry}%
  }%
}
```

Unset all entries for the named glossaries (supplied in a comma-separated list). Syntax: `\glsunsetall[<glossary-list>]`

```

\glsunsetall
  \newcommand*\glsunsetall[1][\@glo@types]{%
    \forallglsentries[#1]{\@glsentry}%
    {%
      \glsunset{\@glsentry}%
    }%
  }

```

As above, but with only a local effect:

```

\glslocalunsetall
  \newcommand*\glslocalunsetall[1][\@glo@types]{%
    \forallglsentries[#1]{\@glsentry}%
    {%
      \glslocalunset{\@glsentry}%
    }%
  }

```

2.9 Keeping Track of How Many Times an Entry Has Been Unset

Version 4.14 introduced `\glsenableentrycount` that keeps track of how many times an entry is marked as used. The counter is reset back to zero when the first use flag is reset. Note that although the word “counter” is used here, it’s not an actual L^AT_EX counter or even an explicit T_EX count register but is just a macro. Any of the commands that use `\glsunset` or `\glslocalunset`, such as `\gls`, will automatically increment this value. Commands that don’t modify the first use flag (such as `\glstext` or `\glsentrytext`) don’t modify this value.

`\newglossaryentry@defcounters` Define entry fields to keep track of how many times that entry has been marked as used.

```

\newcommand*\@newglossaryentry@defcounters{%
  \csdef{glo@\@glo@label @currcount}{0}%
  \csdef{glo@\@glo@label @prevcount}{0}%
}

```

`\ifglsresetcurrcount` A conditional to determine whether or not to reset the entry count to zero if the first use flag is reset. The default is now not to do this, which changes the implementation for v4.49 and below.

```

\newif\ifglsresetcurrcount
\glsresetcurrcountfalse

```

`\glsenableentrycount` Enables tracking of how many times an entry has been marked as used.

```

\newcommand*\glsenableentrycount{%

```

Enable new entry fields.

```

\let\@newglossaryentry@defcounters\@newglossaryentry@defcounters

```

Disable `\newglossaryentry` in the document environment.

```
\renewcommand*\gls@defdocnewglossaryentry}{%
\renewcommand*\newglossaryentry[2]{%
\PackageError{glossaries}{\string\newglossaryentry\space
may only be used in the preamble when entry counting has
been activated}{If you use \string\glsenableentrycount\space
you must place all entry definitions in the preamble not in
the document environment}%
}%
}%
```

Define commands `\glsentrycurrcount` and `\glsentryprevcount` to access these new fields. Default to zero if undefined.

```
\newcommand*\glsentrycurrcount}[1]{%
\ifcsundef{gls@glsdetoklabel{##1}@currcount}%
{0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*\glsentryprevcount}[1]{%
\ifcsundef{gls@glsdetoklabel{##1}@prevcount}%
{0}{\@gls@entry@field{##1}{prevcount}}%
}%
```

Make the `unset` and `reset` functions also increment or reset the entry counter.

```
\renewcommand*\@glsunset}[1]{%
\@glsunset{##1}%
\@gls@increment@currcount{##1}%
}%
\renewcommand*\@glslocalunset}[1]{%
\@glslocalunset{##1}%
\@gls@local@increment@currcount{##1}%
}%
\renewcommand*\@glsreset}[1]{%
\@glsreset{##1}%
\ifglsresetcurrcount\csgdef{gls@glsdetoklabel{##1}@currcount}{0}\fi
}%
\renewcommand*\@glslocalreset}[1]{%
\@glslocalreset{##1}%
\ifglsresetcurrcount\csdef{gls@glsdetoklabel{##1}@currcount}{0}\fi
}%
```

Alter behaviour of `\cgl`s. (Only global `unset` is used if previous count was one as it doesn't make sense to have a local `unset` here given that the previous count was global.)

```
\def\@cgl#1##2[##3]{%
\ifnum\glsentryprevcount{##2}=1\relax
\cglformat{##2}{##3}%
\glsunset{##2}%
\else
\@gls@{##1}{##2}[##3]%
\fi
}%
```


Similarly for the analogous commands. No case change plural:

```
\def\cglsp1@##1##2[##3]{%
  \ifnum\glentryprevcount{##2}=1\relax
    \cglsp1format{##2}{##3}%
    \glunset{##2}%
  \else
    \@glsp1@{##1}{##2}[##3]%
  \fi
}%
```

First letter uppercase singular:

```
\def@cGls@##1##2[##3]{%
  \ifnum\glentryprevcount{##2}=1\relax
    \cGlsformat{##2}{##3}%
    \glunset{##2}%
  \else
    \@Gls@{##1}{##2}[##3]%
  \fi
}%
```

First letter uppercase plural:

```
\def@cGlspl@##1##2[##3]{%
  \ifnum\glentryprevcount{##2}=1\relax
    \cGlsplformat{##2}{##3}%
    \glunset{##2}%
  \else
    \@Glspl@{##1}{##2}[##3]%
  \fi
}%
```

Write information to aux file at the end of the document

```
\AtEndDocument{\@gls@write@entrycounts}%
```

Fetch previous count information from aux file. (No check here to determine if the entry is still defined.)

```
\renewcommand*{\@gls@entry@count}[2]{%
  \csgdef{gls@glsetoklabel{##1}@prevcount}{##2}%
}%
```

`\glsenableentrycount` may only be used once and only in the preamble.

```
\let\glsenableentrycount\relax
}
\@onlypreamble\glsenableentrycount
```

```
\@gls@increment@currcount
```

```
\newcommand*{\@gls@increment@currcount}[1]{%
  \csxdef{gls@glsetoklabel{##1}@currcount}{%
    \number\numexpr\glentrycurrcount{##1}+1}%
}
```

```
s@local@increment@currcount
```

```

\newcommand*{\@gls@local@increment@currcount}[1]{%
  \csedef{glo@glstoklabel{#1}@currcount}{%
    \number\numexpr\glsentrycurrcount{#1}+1}%
  }

```

`\@gls@write@entrycounts` Write the entry counts to the aux file. Use `\immediate` since this occurs right at the end of the document. Only write information for entries that have been used. (Some users have a file containing vast numbers of entries, many of which may not be used. There's no point writing information about the entries that haven't been used and it will only slow things down.)

```

\newcommand*{\@gls@write@entrycounts}{%
  \immediate\write\@auxout
  {\string\providecommand*{\string@gls@entry@count}[2]{}}%
  \forallglsentries{\@glsentry}{%
    \ifglsused{\@glsentry}%
    {\immediate\write\@auxout
      {\string@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}}%
    }%
  }

```

`\@gls@entry@count` Default behaviour is to ignore arguments. Activated by `\glsenableentrycount`.

```

\newcommand*{\@gls@entry@count}[2]{}

```

`\cgl` Define command that works like `\gls` but behaves differently if the entry count function is enabled. (If not enabled, it behaves the same as `\gls` but issues a warning.)

```

\newrobustcmd*{\cgl}{\@gls@hyp@opt@cgl}

```

`\@cgl` Defined the un-starred form. Need to determine if there is a final optional argument

```

\newcommand*{\@cgl}[2][]{%
  \new@ifnextchar[{\@cgl@{#1}{#2}}{\@cgl@{#1}{#2} []}]%
}

```

`\@cgl@` Read in the final optional argument. This defaults to same behaviour as `\gls` but issues a warning.

```

\def\@cgl@#1#2[#3]{%
  \GlossariesWarning{\string@cgl\space is defaulting to
    \string@gls\space since you haven't enabled entry counting}%
  \@gls@{#1}{#2}[#3]%
}

```

`\cglformat` Format used by `\cgl` if entry only used once on previous run. The first argument is the label, the second argument is the insert text.

```

\newcommand*{\cglformat}[2]{%
  \ifglshaslong{#1}{\glsentrylong{#1}}{\glsentryfirst{#1}}#2%
}

```

`\cGls` Define command that works like `\Gls` but behaves differently if the entry count function is enabled. (If not enabled, it behaves the same as `\Gls` but issues a warning.)

```
\newrobustcmd*{\cGls}{\@gls@hyp@opt\@cGls}
```

Add sentence case mapping.

```
\glsmfuaddmap{\cGls}{\cGls}
```

`\@cGls` Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@cGls}[2] [] {%
  \new@ifnextchar[{\@cGls@{#1}{#2}}{\@cGls@{#1}{#2} []}%
}
```

`\@cGls@` Read in the final optional argument. This defaults to same behaviour as `\Gls` but issues a warning.

```
\def\@cGls@#1#2[#3] {%
  \GlossariesWarning{\string\cGls\space is defaulting to
    \string\Gls\space since you haven't enabled entry counting}%
  \@Gls@{#1}{#2}[#3]%
}
```

`\cGlsformat` Format used by `\cGls` if entry only used once on previous run. The first argument is the label, the second argument is the insert text.

```
\newcommand*{\cGlsformat}[2] {%
  \ifglsashaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}#2%
}
```

`\cglsp1` Define command that works like `\glsp1` but behaves differently if the entry count function is enabled. (If not enabled, it behaves the same as `\glsp1` but issues a warning.)

```
\newrobustcmd*{\cglsp1}{\@gls@hyp@opt\@cglsp1}
```

`\@cglsp1` Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@cglsp1}[2] [] {%
  \new@ifnextchar[{\@cglsp1@{#1}{#2}}{\@cglsp1@{#1}{#2} []}%
}
```

`\@cglsp1@` Read in the final optional argument. This defaults to same behaviour as `\glsp1` but issues a warning.

```
\def\@cglsp1@#1#2[#3] {%
  \GlossariesWarning{\string\cglsp1\space is defaulting to
    \string\glsp1\space since you haven't enabled entry counting}%
  \@glsp1@{#1}{#2}[#3]%
}
```

`\cglspformat` Format used by `\cglsp1` if entry only used once on previous run. The first argument is the label, the second argument is the insert text.

```
\newcommand*{\cglspformat}[2]{%
  \ifglshaslong{#1}{\glentrylongpl{#1}}{\glentryfirstplural{#1}}#2%
}
```

`\Glspl` Define command that works like `\Glspl` but behaves differently if the entry count function is enabled. (If not enabled, it behaves the same as `\Glspl` but issues a warning.)

```
\newrobustcmd*{\Glspl}{\@gls@hyp@opt\@cGlspl}
```

Add sentence case mapping.

```
\glsmfuaddmap{\cglsp1}{\cGlspl}
```

`\@cglsp1` Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@cGlspl}[2][ ]{%
  \new@ifnextchar[{\@cGlspl@{#1}{#2}}{\@cGlspl@{#1}{#2}[ ]}%
}
```

`\@cGlspl@` Read in the final optional argument. This defaults to same behaviour as `\Glspl` but issues a warning.

```
\def\@cGlspl@#1#2[#3]{%
  \GlossariesWarning{\string\cGlspl\space is defaulting to
    \string\Glspl\space since you haven't enabled entry counting}%
  \@Glspl@{#1}{#2}[#3]%
}
```

`\cGlsplformat` Format used by `\cGlspl` if entry only used once on previous run. The first argument is the label, the second argument is the insert text.

```
\newcommand*{\cGlsplformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
}
```

2.10 Loading files containing glossary entries

Glossary entries can be defined in an external file. These external files can contain `\newglossaryentry` and `\newacronym` commands.¹

```
\loadglsentries[<type>]{<filename>}
```

This command will input the file using `\input`. The optional argument specifies to which glossary the entries should be assigned if they haven't used the `type` key. If the optional argument is not specified, the default glossary is used. Only those entries used in the document (via `\glslink`, `\gls`, `\glspl` and

¹and any other valid L^AT_EX code that can be used in the preamble.

uppercase variants or `\glsadd` and `\glsaddall` will appear in the glossary). The mandatory argument is the filename (with or without `.tex` extension).

```
\loadglsentries
  \newcommand*{\loadglsentries}[2][\@gls@default]{%
    \let\@gls@default\glsdefaulttype
    \def\glsdefaulttype{#1}\input{#2}%
    \let\glsdefaulttype\@gls@default
  }
```

`\loadglsentries` can only be used in the preamble:

```
\@onlypreamble{\loadglsentries}
```

2.11 Using glossary entries in the text

Any term that has been defined using `\newglossaryentry` (or `\newacronym`) can be displayed in the text (i.e. outside of the glossary) using one of the commands defined in this section. Unless you use `\glslink`, the way the term appears in the text is determined by `\glsdisplayfirst` (if it is the first time the term has been used) or `\glsdisplay` (for subsequent use). Any formatting commands (such as `\textbf`) is governed by `\glstextformat`. By default this just displays the *link text* “as is”.

```
\glstextformat
  \newcommand*{\glstextformat}[1]{#1}
```

`\glsentryfmt` As from version 3.11a, the way in which an entry is displayed is now governed by `\glsentryfmt`. This doesn't take any arguments. The required information is set by commands like `\gls`. This used to use the now deprecated `\glsdisplay` and `\glsdisplayfirst`. Replaced with `\glsgenentryfmt` in v4.50.

```
\newcommand*{\glsentryfmt}{\glsgenentryfmt}
```

`\glsgenentryfmt` Define a generic format that just uses the first, text, plural or first plural keys (or the custom text) with the insert text appended.

```
\newcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
  }%
```

Plural form

```
\glscapscase
  }%
```

Don't adjust case

```
\ifglsused\glslabel
  }%
```

Subsequent use

```
\glsentryplural{\glslabel}\glsinsert  
}%  
{%
```

First use

```
\glsentryfirstplural{\glslabel}\glsinsert  
}%  
}%  
{%
```

Make first letter upper case

```
\ifglsused\glslabel  
{%
```

Subsequent use.

```
\Glsentryplural{\glslabel}\glsinsert  
}%  
{%
```

First use

```
\Glsentryfirstplural{\glslabel}\glsinsert  
}%  
}%  
{%
```

Make all upper case

```
\ifglsused\glslabel  
{%
```

Subsequent use

```
\glsuppercase{\glsentryplural{\glslabel}\glsinsert}%  
}%  
{%
```

First use

```
\glsuppercase{\glsentryfirstplural{\glslabel}\glsinsert}%  
}%  
}%  
}%  
{%
```

Singular form

```
\glscaps case  
{%
```

Don't adjust case

```
\ifglsused\glslabel  
{%
```

Subsequent use

```
\glsentrytext{\glslabel}\glsinsert  
}%  
{%
```

First use

```
\glsentryfirst{\glslabel}\glsinsert  
}%  
}%  
{%
```

Make first letter upper case

```
\ifglsused\glslabel  
{%
```

Subsequent use

```
\Glsentrytext{\glslabel}\glsinsert  
}%  
{%
```

First use

```
\Glsentryfirst{\glslabel}\glsinsert  
}%  
}%  
{%
```

Make all upper case

```
\ifglsused\glslabel  
{%
```

Subsequent use

```
\glsuppercase{\glsentrytext{\glslabel}\glsinsert}%  
}%  
{%
```

First use

```
\glsuppercase{\glsentryfirst{\glslabel}\glsinsert}%  
}%  
}%  
}%  
}%  
{%
```

Custom text provided in `\glsdisp`. (The insert is most likely to be empty at this point.)

```
\glscustomtext\glsinsert  
}%  
}
```

`\glsgenacfmt` Define a generic acronym format that uses the long and short keys (or their plurals) and `\acrfullformat`, `\firstacronymfont` and `\acronymfont`.

```
\newcommand*{\glsgenacfmt}{%  
  \ifdefempty\glscustomtext  
  {%  
    \ifglsused\glslabel  
    {%
```

Subsequent use:

```
\glsifplural  
{%
```

Subsequent plural form:

```
\glsifplural  
{%
```

Subsequent plural form, don't adjust case:

```
\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert  
}%  
{%
```

Subsequent plural form, make first letter upper case:

```
\acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert  
}%  
{%
```

Subsequent plural form, all caps:

```
\glsupper  
{\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert}%  
}%  
{%
```

Subsequent singular form

```
\glsifplural  
{%
```

Subsequent singular form, don't adjust case:

```
\acronymfont{\glsentryshort{\glslabel}}\glsinsert  
}%  
{%
```

Subsequent singular form, make first letter upper case:

```
\acronymfont{\Glsentryshort{\glslabel}}\glsinsert  
}%  
{%
```

Subsequent singular form, all caps:

```
\glsupper  
{\acronymfont{\glsentryshort{\glslabel}}\glsinsert}%  
}%  
}%  
{%
```

First use:

```
\glsifplural  
{%
```

First use plural form:

```
\glsifplural  
{%
```


First use plural form, don't adjust case:

```
\genplacrfullformat{\glslabel}{\glsinsert}%  
}%  
{%
```

First use plural form, make first letter upper case:

```
\Genplacrfullformat{\glslabel}{\glsinsert}%  
}%  
{%
```

First use plural form, all caps:

```
\glsuppercase  
  {\genplacrfullformat{\glslabel}{\glsinsert}}%  
}%  
}%  
{%
```

First use singular form

```
\glscapscase  
{%
```

First use singular form, don't adjust case:

```
\genacrfullformat{\glslabel}{\glsinsert}%  
}%  
{%
```

First use singular form, make first letter upper case:

```
\Genacrfullformat{\glslabel}{\glsinsert}%  
}%  
{%
```

First use singular form, all caps:

```
\glsuppercase  
  {\genacrfullformat{\glslabel}{\glsinsert}}%  
}%  
}%  
}%  
}%  
{%
```

User supplied text.

```
\glscustomtext  
}%  
}
```

```
\genacrfullformat{<label>}{<insert>}
```

`\genacrfullformat`

The full format used by `\glsngenacfmt` (singular).

```
\newcommand*{\genacrfullformat}[2]{%  
  \glsentrylong{#1}#2\space
```

```
(\protect\firstacronymfont{\glsentryshort{#1}})%
}
```

`\Genacrfullformat`

```
\Genacrfullformat{<label>}{<insert>}
```

As above but makes the first letter upper case.

```
\newcommand*{\Genacrfullformat}[2]{%
  \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}
```

`\genplacrfullformat`

```
\genplacrfullformat{<label>}{<insert>}
```

The full format used by `\glsngenacfmt` (plural).

```
\newcommand*{\genplacrfullformat}[2]{%
  \glsentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glsentryshortpl{#1}})%
}
```

`\Genplacrfullformat`

```
\Genplacrfullformat{<label>}{<insert>}
```

As above but makes the first letter upper case.

```
\newcommand*{\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}
```

`\glsdisplayfirst` `\glsdisplayfirst` deprecated in v3.11a (2013-10-15) and removed in v4.50. Use rollback if required.

`\glsdisplay` `\glsdisplay` deprecated in v3.11a (2013-10-15) and removed in v4.50. Use rollback if required.

`\defglsdisplay` `\defglsdisplay` deprecated in v3.11a (2013-10-15) and removed in v4.50. Use rollback if required.

`\defglsdisplayfirst` `\defglsdisplayfirst` deprecated in v3.11a (2013-10-15) and removed in v4.50. Use rollback if required.

2.11.1 Links to glossary entries

The links to glossary entries all have a first optional argument that can be used to change the format and counter of the associated entry number. Except for `\glslink` and `\glsdisp`, the commands like `\gls` have a final optional argument that can be used to insert additional text in the link (this will

usually be appended, but can be redefined using `\defglsentryfmt`). It goes against the L^AT_EX norm to have an optional argument after the mandatory arguments, but it makes more sense to write, say, `\gls{label}[’s]` rather than, say, `\gls[append=’s]{label}`. Since these control sequences are defined to include the final square bracket, spaces will be ignored after them. This is likely to lead to confusion as most users would not expect, say, `\gls{<label>}` to ignore following spaces, so `\new@ifnextchar` from the `amsgen` package is required.

The following keys can be used in the first optional argument. The `counter` key checks that the value is the name of a valid counter.

```
\define@key{glslink}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
  \def\@gls@counter{#1}%
}%
}
```

The value of the `format` key should be the name of a command (without the initial backslash) that has a single mandatory argument which can be used to format the associated entry number.

```
\define@key{glslink}{format}{%
  \def\@glsnumberformat{#1}}
```

`s@glslink@hyper@update@hook`

```
\newcommand*{\@gls@glslink@hyper@update@hook}{}
```

The `hyper` key is a boolean key, it can either have the value `true` or `false`, and indicates whether or not to make a hyperlink to the relevant glossary entry. If `hyper` is `false`, an entry will still be made in the glossary, but the given text won’t be a hyperlink.

```
\define@boolkey{glslink}{hyper}[true]{\@gls@glslink@hyper@update@hook}
```

Initialise hyper key.

```
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
```

Commands like `\gls` and `\glsdisp` unset the first use flag afterwards. Provide a way to localise this.

`\@gls@save@glslocal`

```
\newcommand*{\@gls@save@glslocal}{%
  \let\if@org@KV@glslink@local\ifKV@glslink@local
}
```

```

\@gls@restore@glslocal
\newcommand*{\@gls@restore@glslocal}{%
  \ifKV@glslink@local
    \let\@gls@do@glsunset@glslocalunset
  \else
    \let\@gls@do@glsunset@glsunset
  \fi
  \let\ifKV@glslink@local\if@org@KV@glslink@local
}

```

```

\@gls@do@glsunset
\newcommand*{\@gls@do@glsunset}[1]{\glsunset{#1}}

```

The local key is a boolean key. If true this indicates that commands such as `\gls` should only do a local reset rather than a global one.

```

\define@boolkey{glslink}{local}[true]{}
\@gls@save@glslocal

```

The original `\glsifhyper` command isn't particularly useful as it makes more sense to check the actual hyperlink setting rather than testing whether the starred or unstarred version has been used. Therefore, as from version 4.08, `\glsifhyper` is deprecated in favour of `\glsifhyperon`. In case there is a particular need to know whether the starred or unstarred version was used, provide a new command that determines whether the *-version, +-version or unmodified version was used.

```

\glslinkvar{<unmodified case>}{<star case>}{<plus case>}

```

`\glslinkvar` Initialise to unmodified case.

```

\newcommand*{\glslinkvar}[3]{#1}

```

`\glsifhyper` `\glsifhyper` deprecated in v4.08 (2014-07-30) and removed in v4.50.

`\@gls@hyp@opt` Used by the commands such as `\glslink` to determine whether to modify the hyper option.

```

\newcommand*{\@gls@hyp@opt}[1]{%
  \let\glslinkvar\@firstofthree
  \let\@gls@hyp@opt@cs#1\relax
  \ifstar{\s@gls@hyp@opt}%
  {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}

```

`\s@gls@hyp@opt` Starred version

```

\newcommand*{\s@gls@hyp@opt}[1] []{%
  \let\glslinkvar\@secondofthree
  \@gls@hyp@opt@cs[hyper=false,#1]}

```

`\p@gls@hyp@opt` Plus version

```
\newcommand*\p@gls@hyp@opt}[1] [] {%  
  \let\glslinkvar\@thirdofthree  
  \@gls@hyp@opt@cs[hyper=true,#1]}
```

Syntax:

```
\glslink[<options>]{<label>}{<text>}
```

Display *<text>* in the document, and add the entry information for *<label>* into the relevant glossary. The optional argument should be a key value list using the `glslink` keys defined above.

There is also a starred version:

```
\glslink*[<options>]{<label>}{<text>}
```

which is equivalent to `\glslink[hyper=false,<options>]{<label>}{<text>}`

First determine which version is being used:

`\glslink`

```
\newrobustcmd*\glslink}{%  
  \@gls@hyp@opt\@gls@link  
}
```

`\@gls@link` The main part of the business is in `\@gls@link` which shouldn't check if the term is defined as it's called by `\gls` etc which also perform that check.

```
\newcommand*\@gls@link}[3] [] {%  
  \glsdoifexistsordo{#2}%  
  {%  
    \let\do@gls@link@checkfirsthyper\relax  
    \@gls@link[#1]{#2}{#3}%  
  }{%  
  }
```

Display the specified text. (The entry doesn't exist so there's nothing to link it to.)

```
  \glstextformat{#3}%  
}%  
  
  \glspostlinkhook  
}
```

`\Glslink` Provide a sentence-case version in order to set up a map.

```
\newrobustcmd*\Glslink}{%  
  \@gls@hyp@opt\@Gls@link  
}
```

```

\newcommand*{\@Gls@link}[3][\%
  \@gls@link[#1]{#2}{\glsentencecase{#3}}%
}
\glsmfuaddmap{\glslink}{\Glslink}

\glspostlinkhook
\newcommand*{\glspostlinkhook}{}

\@gls@link@checkfirsthyper Check for first use and switch off hyper key if hyperlink not wanted. (Should be
off if first use and hyper=false is on or if first use and both the entry is in an
acronym list and the acrfootnote setting is on.) This assumes the glossary type
is stored in \glsstype and the label is stored in \glslabel.
\newcommand*{\@gls@link@checkfirsthyper}{%
  \ifglsused{\glslabel}%
  {%
  }%
  {%
    \gls@checkisacronymlist\glsstype
    \ifglshyperfirst
      \if@glsisacronymlist
        \ifglsacrfootnote
          \KV@glslink@hyperfalse
        \fi
      \fi
    \else
      \KV@glslink@hyperfalse
    \fi
  }%
}
Allow user to hook into this
\glslinkcheckfirsthyperhook
}

\glslinkcheckfirsthyperhook Allow user to hook into the \@gls@link@checkfirsthyper macro
\newcommand*{\glslinkcheckfirsthyperhook}{}

\glslinkpostsetkeys
\newcommand*{\glslinkpostsetkeys}{}

\glsifhyperon Check the value of the hyper key:
\newcommand{\glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}

\do@glsdisablehyperinlist Disable hyperlink if in the “nohyper” list.
\newcommand*{\do@glsdisablehyperinlist}{%
  \expandafter\DTLifinlist\expandafter{\glsstype}{\@gls@nohyperlist}%
  {\KV@glslink@hyperfalse}{}%
}

\gls@setdefault@glslink@opts Hook to set default options for \@glslink.
\newcommand*{\gls@setdefault@glslink@opts}{}

```

```

\@gls@link
  \def\@gls@link[#1]#2#3{%
Inserting \leavevmode suggested by Donald Arseneau (avoids problem with
tabularx).
  \leavevmode
  \protected@edef\glslabel{\glsdetoklabel{#2}}%
Save options in \@gls@link@opts and label in \@gls@link@label
  \def\@gls@link@opts{#1}%
  \let\@gls@link@label\glslabel

  \def\@glsnumberformat{glsnumberformat}%
  \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%

If this is in one of the “nohypertypes” glossaries, suppress the hyperlink by
default
  \protected@edef\glstype{\csname glo@\glslabel @type\endcsname}%
Save original setting
  \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
Save original setting of local key.
  \@gls@save@glslocal

Set defaults:
  \@gls@setdefault@glslink@opts
Switch off hyper setting if the glossary type has been identified in nohyperlist.
  \do@glsdisablehyperinlist

Macros must set this before calling \@gls@link. The commands that check
the first use flag should set this to \@gls@link@checkfirsthyper otherwise it
should be set to \relax.
  \do@gls@link@checkfirsthyper
  \setkeys{glslink}{#1}%

Add a hook for the user to customise things after the keys have been set.
  \glslinkpostsetkeys

Store the entry’s counter in \theglsentrycounter
  \@gls@saveentrycounter

Define sort key if necessary:
  \@gls@setsort{\glslabel}%
(De-tok’ing done by \@do@wrglossary)
  \glsencapwrcontent{\@do@wrglossary{#2}}%
  \ifKV@glslink@hyper
    \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
  \else
    \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
  \fi

```

Restore original setting

```
\let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
\@gls@restore@glslocal
}
```

`\glsencapwrcontent` This command is also provided by glossaries-extra v1.49. It's purpose is to resolve [issue #189](#).

```
\newcommand*{\glsencapwrcontent}[1]{\#1}
```

`\glolinkprefix`

```
\newcommand*{\glolinkprefix}{glo:}
```

`\glsentrycounter` Set default value of entry counter

```
\def\glsentrycounter{\glscounter}%
```

`\@gls@saveentrycounter` Need to check if using equation counter in align environment:

```
\newcommand*{\@gls@saveentrycounter}{%
\def\@gls@Hcounter{}}
```

Are we using equation counter?

```
\ifthenelse{\equal{\@gls@counter}{equation}}{%
{
```

If we're in align environment, `\xatlevel@` will be defined. (Can't test for `\@currenvir` as may be inside an inner environment.)

```
\ifcsundef{xatlevel@}%
{%
\edef\theglsentrycounter{\expandafter\noexpand
\csname the\@gls@counter\endcsname}%
}%
{%
\ifx\xatlevel@\@empty
\edef\theglsentrycounter{\expandafter\noexpand
\csname the\@gls@counter\endcsname}%
\else
\savecounters@
\advance\c@equation by 1\relax
\edef\theglsentrycounter{\csname the\@gls@counter\endcsname}%
}
```

Check if hyperref version of this counter

```
\ifcsundef{theH\@gls@counter}%
{%
\def\@gls@Hcounter{\theglsentrycounter}%
}%
{%
\def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
}%
\protected@edef\theglsentrycounter{\@gls@Hcounter}%
\restorecounters@
\fi
```



```

    }%
  }%
  {%

```

Not using equation counter so no special measures:

```

\edef\theglentrycounter{\expandafter\noexpand
\csname the\@gls@counter\endcsname}%
}%

```

Check if hyperref version of this counter

```

\ifx\@gls@Hcounter\@empty
\ifcsundef{theH\@gls@counter}%
{%
\def\theHglentrycounter{\theglentrycounter}%
}%
{%
\protected@edef\theHglentrycounter{\expandafter\noexpand
\csname theH\@gls@counter\endcsname}%
}%
\fi
}

```

`\@set@glo@numformat` Set the formatting information in the format required by `makeindex`. The first argument is the format specified by the user (via the format key), the second argument is the name of the counter used to indicate the location, the third argument is a control sequence which stores the required format and the fourth argument (new to v3.0) is the hyper-prefix.

```

\def\@set@glo@numformat#1#2#3#4{%
\expandafter\@glo@check@mkidxrangear#3\@nil
\protected@edef#1{%
\@glo@prefix setentrycounter[#4]{#2}%
\expandafter\string\csname\@glo@suffix\endcsname
}%
\@gls@checkmkidxchars#1%
}

```

Check to see if the given string starts with a (or). If it does set `\@glo@prefix` to the starting character, and `\@glo@suffix` to the rest (or `glsnumberformat` if there is nothing else), otherwise set `\@glo@prefix` to nothing and `\@glo@suffix` to all of it.

```

\def\@glo@check@mkidxrangear#1#2\@nil{%
\if#1(\relax
\def\@glo@prefix{()}%
\if\relax#2\relax
\def\@glo@suffix{glsnumberformat}%
\else
\def\@glo@suffix{#2}%
\fi
\else

```

```

\if#1)\relax
\def\@glo@prefix{)}%
\if\relax#2\relax
\def\@glo@suffi{x{glsnumberformat}}%
\else
\def\@glo@suffi{x{#2}}%
\fi
\else
\def\@glo@prefix{}\def\@glo@suffi{x{#1#2}}%
\fi
\fi}

```

`\@gls@escbsdq` Escape backslashes and double quote marks. The argument must be a control sequence.

```

\newcommand*{\@gls@escbsdq}[1]{%
\def\@gls@checkedmkidx{}%
\let\gls@xdystring=#1\relax
\@onelevel@sanitize\gls@xdystring
\edef\do@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\@null}%
\do@gls@xdycheckbackslash
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\@null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%

```

Unsanitize `\gls@numberpage`, `\gls@alphpage`, `\gls@Alphpage` and `\gls@romanpage` (thanks to David Carlisle for the suggestion.) Skip if `esclocations=false`.

```

\ifglsclocations
\for\@gls@tmp:=\gls@protected@pagefmts\do
{%
\edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\ \expandonce\@gls@tmp}%
\@onelevel@sanitize\@gls@sanitized@tmp
\edef\gls@dostubst{%
\noexpand\DTLsubstituteall\noexpand\gls@xdystring
{\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
}%
\gls@dostubst
}%
\fi

```

Assign to required control sequence

```

\let#1=\gls@xdystring
}

```

Catch special characters (argument must be a control sequence):

```

\@gls@checkmkidxchars
\newcommand{\@gls@checkmkidxchars}[1]{%

```

```

\ifglxindy
  \@gls@escbsdq{#1}%
\else
  \def\@gls@checkedmkidx{%
  \expandafter\@gls@checkquote#1\@nil""\null
  \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
  \def\@gls@checkedmkidx{%
  \expandafter\@gls@checkescquote#1\@nil""\null
  \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
  \def\@gls@checkedmkidx{%
  \expandafter\@gls@checkescactual#1\@nil"??\null
  \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
  \def\@gls@checkedmkidx{%
  \expandafter\@gls@checkactual#1\@nil??\null
  \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
  \def\@gls@checkedmkidx{%
  \expandafter\@gls@checkbar#1\@nil||\null
  \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
  \def\@gls@checkedmkidx{%
  \expandafter\@gls@checkescbar#1\@nil|||\null
  \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
  \def\@gls@checkedmkidx{%
  \expandafter\@gls@checklevel#1\@nil!!\null
  \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%

  \def\@gls@checkedmkidx{%
  \expandafter\@gls@checkesclevel#1\@nil!|\!|\null
  \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
  \def\@gls@checkedmkidx{%
  \edef\@do@gls@checkesc{%
    \noexpand\@gls@checkesc\expandonce{#1}\noexpand\@nil
    \@backslashchar\@backslashchar\noexpand\@nil
  }%
  \@do@gls@checkesc
  \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
  \fi
}
}

```

Update the control sequence and strip trailing \@nil:

```

\@gls@updatechecked
  \def\@gls@updatechecked#1\@nil#2{\def#2{#1}}

\@gls@tmpb Define temporary token
  \newtoks\@gls@tmpb

\@gls@checkquote Replace " with "" since " is a makeindex special character.
  \def\@gls@checkquote#1"#2"#3\@nil{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@=#1}%

```

```

\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
\def\@gls@checkquote{\@gls@checkquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@quotechar}%
\ifx\null#3\null
\def\@gls@checkquote{\@gls@checkquote#2""\null}%
\else
\def\@gls@checkquote{\@gls@checkquote#2"#3\null}%
\fi
\fi
\@gls@checkquote
}

```

\@gls@checkescquote Do the same for \":

```

\def\@gls@checkescquote#1\#2\#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\@\@gls@quotechar
\@gls@quotechar\string\@\@gls@quotechar}%
\def\@gls@checkescquote{\@gls@checkescquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\@\@gls@quotechar}%
\ifx\null#3\null
\def\@gls@checkescquote{\@gls@checkescquote#2\""\null}%
\else
\def\@gls@checkescquote{\@gls@checkescquote#2\#3\null}%
\fi
\fi
\@gls@checkescquote
}

```

\@gls@checkescactual Similarly for \? (which is replaces @ as makeindex's special character):

```

\def\@gls@checkescactual#1\?#2\?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%

```

```

\toks@={#1}%
\ifx\null#2\null
  \ifx\null#3\null
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
    \def\@gls@checkescactual{\relax}%
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar
\@gls@quotechar\string"\@gls@actualchar}%
    \def\@gls@checkescactual{\@gls@checkescactual#3\null}%
  \fi
\else
  \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar}%
  \ifx\null#3\null
    \def\@gls@checkescactual{\@gls@checkescactual#2\?\?\null}%
  \else
    \def\@gls@checkescactual{\@gls@checkescactual#2\?#3\null}%
  \fi
\fi
\@gls@checkescactual
}

```

\@gls@checkescbar Similarly for \|:

```

\def\@gls@checkescbar#1\|#2\|#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkescbar{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@encapchar
\@gls@quotechar\string"\@gls@encapchar}%
      \def\@gls@checkescbar{\@gls@checkescbar#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@encapchar}%
    \ifx\null#3\null
      \def\@gls@checkescbar{\@gls@checkescbar#2\|\|\null}%
    \else
      \def\@gls@checkescbar{\@gls@checkescbar#2\|#3\null}%
    \fi
  \fi
\@gls@checkescbar
}

```

\@gls@checkesclevel Similarly for \!:

```

\def\@gls@checkesclevel#1\!#2\!#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkesclevel{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\string"\@gls@levelchar
        \@gls@quotechar\string"\@gls@levelchar}%
      \def\@gls@checkesclevel{\@gls@checkesclevel#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\string"\@gls@levelchar}%
    \ifx\null#3\null
      \def\@gls@checkesclevel{\@gls@checkesclevel#2\!\!\null}%
    \else
      \def\@gls@checkesclevel{\@gls@checkesclevel#2\!#3\null}%
    \fi
  \fi
  \@gls@checkesclevel
}

```

\@gls@checkbar and for |:

```

\def\@gls@checkbar#1|#2|#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkbar{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
      \def\@gls@checkbar{\@gls@checkbar#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@gls@encapchar}%
    \ifx\null#3\null
      \def\@gls@checkbar{\@gls@checkbar#2||\null}%
    \else
      \def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
    \fi
  \fi
  \@gls@checkbar
}

```

\@gls@checklevel and for !:

```
\def\@gls@checklevel#1!#2!#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@@gls@checklevel{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
      \def\@@gls@checklevel{\@gls@checklevel#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@gls@levelchar}%
    \ifx\null#3\null
      \def\@@gls@checklevel{\@gls@checklevel#2!\null}%
    \else
      \def\@@gls@checklevel{\@gls@checklevel#2!#3\null}%
    \fi
  \fi
  \@@gls@checklevel
}
```

\@gls@checkactual and for ?:

```
\def\@gls@checkactual#1?#2?#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@@gls@checkactual{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
      \def\@@gls@checkactual{\@gls@checkactual#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@gls@actualchar}%
    \ifx\null#3\null
      \def\@@gls@checkactual{\@gls@checkactual#2??\null}%
    \else
      \def\@@gls@checkactual{\@gls@checkactual#2?#3\null}%
    \fi
  \fi
  \@@gls@checkactual
}
```

\@gls@checkesc and for \:

```
\bgroup
\catcode'\|0\relax
\catcode'\|12\relax
\gdef|\@gls@checkesc#1\#2\#3|null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx|null#2|null
    \ifx|null#3|null
      \edef|\@gls@checkedmkidx{\the|\@gls@tmpb\the|\toks@}%
      \def|\@gls@checkesc{\relax}%
    \else
      \edef|\@gls@checkedmkidx{\the|\@gls@tmpb\the|\toks@
        \@gls@quotechar|\backslashchar|\@gls@quotechar|\backslashchar}%
      \def|\@gls@checkesc{\@gls@checkesc#3|null}%
    \fi
  \else
    \edef|\@gls@checkedmkidx{\the|\@gls@tmpb\the|\toks@
      \@gls@quotechar|\backslashchar}%
    \ifx|null#3|null
      \def|\@gls@checkesc{\@gls@checkesc#2\|\null}%
    \else
      \def|\@gls@checkesc{\@gls@checkesc#2\#3|null}%
    \fi
  \fi
  |\@gls@checkesc
}
\egroup
```

\@gls@xdycheckquote As before but for use with xindy

```
\def|\@gls@xdycheckquote#1"#2"#3|null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx|null#2|null
    \ifx|null#3|null
      \edef|\@gls@checkedmkidx{\the|\@gls@tmpb\the|\toks@}%
      \def|\@gls@xdycheckquote{\relax}%
    \else
      \edef|\@gls@checkedmkidx{\the|\@gls@tmpb\the|\toks@
        \string"\string"}%
      \def|\@gls@xdycheckquote{\@gls@xdycheckquote#3|null}%
    \fi
  \else
    \edef|\@gls@checkedmkidx{\the|\@gls@tmpb\the|\toks@
      \string"}%
    \ifx|null#3|null
      \def|\@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
    \else
      \def|\@gls@xdycheckquote{\@gls@xdycheckquote#2"#3|null}%
    \fi
}
```



```

    \fi
  \@gls@xdycheckquote
}

```

`\@gls@xdycheckbackslash` Need to escape all backslashes for xindy. Define command that will define

```

\@gls@xdycheckbackslash
\edef\def@gls@xdycheckbackslash{%
\noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
##2\@backslashchar##3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter
{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={##1}%
\noexpand\ifx\noexpand\null##2\noexpand\null
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar}%
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
\@backslashchar\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
##3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@xdycheckbackslash
}%
}

```

Now go ahead and define `\@gls@xdycheckbackslash`

```

\def@gls@xdycheckbackslash

```

`\glsdohypertarget`

```

\newlength@gls@tmplen
\newcommand*{\glsdohypertarget}[2]{%
\glsdoshowtarget{#1}%
}%

```

```

        \glsmeasureheight{\gls@tmplen}{#2}%
        \glsdohypertargethook{#1}{#2}%
        \raisebox{\gls@tmplen}{\hypertarget{#1}{}}%
        #2%
    }%
}

```

```

\glsdohypertargethook Hook for \glsdohypertarget
    \newcommand*\glsdohypertargethook[2]{}

```

`\glslabelhypertarget` Provided for use in the definition of `\glsdohypertargethook` if required.

```

\NewDocumentCommand\glslabelhypertarget{mm}{%
\protected@write\@auxout
{\glslabelhypertargetdefs}%
{%
\string\newlabel
{\glslabelhypertargetprefix#1}%
{%
\glslabelhypertargetvalue}{\thepage}{#2}{#1}{}%
}%
}%
}

```

```

\ExplSyntaxOn

```

`\glslabelhypertargetvalue`

```

\newcommand\glslabelhypertargetvalue{
\bool_lazy_and:nnT
{ \tl_if_exist_p:N \glscurrententrylabel }
{ \bool_not_p:n { \tl_if_empty_p:N \glscurrententrylabel } }
{
\glsentryname \glscurrententrylabel
}
}
\ExplSyntaxOff

```

`\glslabelhypertargetprefix`

```

\newcommand*\glslabelhypertargetprefix{}

```

`\glslabelhypertargetdefs`

```

\newcommand*\glslabelhypertargetdefs{%
\let\glossentryname\glsentryname
\let\glossentrydesc\glsentrydesc
\let\glossentrysymbol\glsentrysymbol
\let\Glossentryname\glsentryname
\let\Glossentrydesc\glsentrydesc
\let\Glossentrysymbol\glsentrysymbol
\let\glossentrynameother\@gls@entry@field
}

```

```

\glsdohyperlink
  \newcommand*{\glsdohyperlink}[2]{%
    \glsdoshowtarget{#1}%
    {%
      \glsdohyperlinkhook{#1}{#2}%
      \hyperlink{#1}{#2}%
    }%
  }

```

```

\glsdohyperlinkhook Hook for \glsdohyperlink
  \newcommand*{\glsdohyperlinkhook}[2]{}

```

```

\glsdonohyperlink
  \newcommand*{\glsdonohyperlink}[2]{#2}

```

`\@glslink` If `\hyperlink` is not defined `\@glslink` ignores its first argument and just does the second argument, otherwise it is equivalent to `\hyperlink`.

```

\ifcsundef{hyperlink}%
{%
  \let\@glslink\glsdonohyperlink
}%
{%
  \let\@glslink\glsdohyperlink
}

```

`\@glstarget` If `\hypertarget` is not defined, `\@glstarget` ignores its first argument and just does the second argument, otherwise it is equivalent to `\hypertarget`.

```

\ifcsundef{hypertarget}%
{%
  \let\@glstarget\@secondoftwo
}%
{%
  \let\@glstarget\glsdohypertarget
}

```

Glossary hyperlinks can be disabled using `\glsdisablehyper` (effect can be localised):

```

\glsdisablehyper
  \newcommand{\glsdisablehyper}{%
    \KV@glslink@hyperfalse
    \let\@glslink\glsdonohyperlink
    \let\@glstarget\@secondoftwo
  }

```

Glossary hyperlinks can be enabled using `\glsenablehyper` (effect can be localised):

`\glsenablehyper`

```
\newcommand{\glsenablehyper}{%
  \KV@glslink@hypertrue
  \let\@glslink\glsdohyperlink
  \let\@gls@target\glsdohypertarget
}
```

Provide some convenience commands if not already defined:

```
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
```

Syntax:

```
\gls[options]{label}[insert text]
```

Link to glossary entry using singular form. The link text is taken from the value of the text or first keys used when the entry was defined.

The first optional argument is a key-value list, the same as `\glslink`, the mandatory argument is the entry label. After the mandatory argument, there is another optional argument to insert extra text in the link text (the location of the inserted text is governed by `\glsdisplay` and `\glsdisplayfirst`). As with `\glslink` there is a starred version which is the same as the unstarred version but with the `hyper` key set to `false`. (Additional options can also be specified in the first optional argument.)

First determine which version is being used:

`\gls`

```
\newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}
```

Defined the un-starred form. Need to determine if there is a final optional argument

`\@gls`

```
\newcommand*{\@gls}[2][ ]{%
  \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2}[]}%
}
```

`\@gls@` Read in the final optional argument:

```
\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper

    \let\glsifplural\@secondoftwo
    \let\gls@scapscase\@firstofthree
    \let\gls@customtext\@empty
  }
  \def\glsinsert{#3}%
}
```

Determine what the link text should be (this is stored in `\@glo@text`) Note that `\@gls@link` sets `\glstype`.

```
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
```

Call `\@gls@link`. If footnote package option has been used and the glossary type is `\acronymtype`, suppress hyperlink for first use. Likewise if the `hyperfirst=false` package option is used.

```
\@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
\@gls@do@glsunset{#2}%
}%

\glspostlinkhook
}
```

`\Gls` behaves like `\gls`, but the first letter of the link text is converted to uppercase (note that if the first letter has an accent, the accented letter will need to be grouped when you define the entry). It is mainly intended for terms that start a sentence:

`\Gls`

```
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
```

Add sentence case mapping.

```
\glsmfuaddmap{\gls}{\Gls}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Gls}[2] [] {%
  \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2} [] }%
}
```

`\@Gls@` Read in the final optional argument:

```
\def\@Gls@#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper

    \let\glsifplural\@secondoftwo
    \let\glsifcaps\@secondofthree
    \let\glsifcustomtext\@empty
  }
  \def\glsinsert{#3}%
}
```

Determine what the link text should be (this is stored in `\@glo@text`) Note that `\@gls@link` sets `\glstype`.

```
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
```

Call `\@gls@link` If footnote package option has been used and the glossary type is `\acronymtype`, suppress hyperlink for first use. Likewise if the `hyperfirst=false` package option is used.

```
\@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
\@gls@do@glsunset{#2}%
}%

\glspostlinkhook
}
```

\GLS behaves like \gls, but the link text is converted to uppercase:

```
\GLS
\newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}
\glsmfublocker{\GLS}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\GLS}[2][ ]{%
  \new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2}[ ]}%
}
```

\@GLS@ Read in the final optional argument:

```
\def\@GLS@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper

    \let\glsifplural\@secondoftwo
    \let\glsupcase\@thirdofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
  }
```

Determine what the link text should be (this is stored in \@glo@text). Note that \@gls@link sets \gls@type.

```
\def\@glo@text{\csname gls@\gls@type @entryfmt@endcsname}%
```

Call \@gls@link If footnote package option has been used and the glossary type is \acronym@type, suppress hyperlink for first use. Likewise if the hyperfirst=false package option is used.

```
\@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
\@gls@do@glsunset{#2}%
}%

\glspostlinkhook
}
```

\glspl behaves in the same way as \gls except it uses the plural form.

```
\glspl
\newrobustcmd*{\glspl}{\@gls@hyp@opt\@glspl}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glspl}[2] [] {%
  \new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2} []}%
}
```

`\@glspl@` Read in the final optional argument:

```
\def\@glspl@#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper

    \let\glsifplural\@firstoftwo
    \let\glsescapscase\@firstofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%

```

Determine what the link text should be (this is stored in `\@glo@text`) Note that `\@gls@link` sets `\glstype`.

```
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
```

Call `\@gls@link`. If footnote package option has been used and the glossary type is `\acronymtype`, suppress hyperlink for first use. Likewise if the `hyperfirst=false` package option is used.

```
\@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
\@gls@do@glsunset{#2}%
}%
```

```
\glspostlinkhook
}
```

`\Glspl` behaves in the same way as `\glspl`, except that the first letter of the link text is converted to uppercase (as with `\Gls`, if the first letter has an accent, it will need to be grouped).

`\Glspl`

```
\newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glspl}{\Glspl}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glspl}[2] [] {%
  \new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2} []}%
}
```

`\@Glspl@` Read in the final optional argument:

```
\def\@Glspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper

    \let\glsifplural\@firstoftwo
    \let\gls caps case\@secondofthree
    \let\gls custom text\@empty
    \def\glsinsert{#3}%
```

Determine what the link text should be (this is stored in `\@glo@text`). This needs to be expanded so that the `\@glo@text` can be passed to `\xmakefirstuc`. Note that `\@gls@link` sets `\glstype`.

```
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
```

Call `\@gls@link`. If footnote package option has been used and the glossary type is `\acronym type`, suppress hyperlink for first use. Likewise if the `hyperfirst=false` package option is used.

```
\@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
\@gls@do@glsunset{#2}%
}%

\gls post link hook
}
```

`\GLSp1` behaves like `\glspl` except that all the link text is converted to uppercase.

`\GLSp1`

```
\newrobustcmd*{\GLSp1}{\@gls@hyp@opt\@GLSp1}
\glsmfublocker{\GLSp1}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSp1}[2][ ]{%
  \new@ifnextchar[{\@GLSp1@{#1}{#2}}{\@GLSp1@{#1}{#2}[ ]}%
}
```

`\@GLSp1` Read in the final optional argument:

```
\def\@GLSp1@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper

    \let\glsifplural\@firstoftwo
    \let\gls caps case\@thirdofthree
    \let\gls custom text\@empty
    \def\glsinsert{#3}%
```


Determine what the link text should be (this is stored in `\@glo@text`) Note that `\@gls@link` sets `\glstype`.

```
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
```

Call `\@gls@link`. If footnote package option has been used and the glossary type is `\acronymtype`, suppress hyperlink for first use. Likewise if the `hyperfirst=false` package option is used.

```
\@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
\@gls@do@glsunset{#2}%
```

```
}%
```

```
\glspostlinkhook
```

```
}
```

`\glsdisp` `\glsdisp[options]{label}{text}` This is like `\gls` except that the link text is provided. This differs from `\glslink` in that it uses `\glsdisplay` or `\glsdisplayfirst` and unsets the first use flag.

First determine if we are using the starred form:

```
\newrobustcmd*\@glsdisp{\@gls@hyp@opt\@glsdisp}
```

Defined the un-starred form.

```
\@glsdisp
```

```
\newcommand*\@glsdisp[3] [] {%
```

```
\glsdoifexists{#2}{%
```

```
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
```

```
\let\glsifplural\@secondoftwo
```

```
\let\glsifscapscase\@firstofthree
```

```
\def\glscustomtext{#3}%
```

```
\def\glsinsert{}%
```

Determine what the link text should be (this is stored in `\@glo@text`) Note that `\@gls@link` sets `\glstype`.

```
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
```

Call `\@gls@link`. If footnote package option has been used and the glossary type is `\acronymtype`, suppress hyperlink for first use. Likewise if the `hyperfirst=false` package option is used.

```
\@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
\@gls@do@glsunset{#2}%
```

```
}%
```

```
\glspostlinkhook
```

```
}
```

`\Glsdisp` Provide a sentence-case version to make it easier to map.
`\newrobustcmd*{\Glsdisp}{\@gls@hyp@opt\@Glsdisp}`

`\@Glsdisp`
`\newcommand*{\@Glsdisp}[3] [] {%`
`\@glsdisp[#1]{#2}{\glsentencecase{#3}}%`
`}`
`\glsmfuaddmap{\glsdisp}{\Glsdisp}`

`\@gls@link@nocheckfirsthyper` Instead of just setting `\do@gls@link@checkfirsthyper` to `\relax` in `\@gls@field@link`, set it to `\@gls@link@nocheckfirsthyper` in case some other action needs to take place.
`\newcommand*{\@gls@link@nocheckfirsthyper}{}`

`\@gls@field@link{<options>}{<label>}{<link text>}`

`\@gls@field@link`

```
\newcommand{\@gls@field@link}[3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \@gls@link[#1]{#2}{#3}%
  }%

  \glspostlinkhook
}
```

`\gls` behaves like `\gls` except it always uses the value given by the text key and it doesn't mark the entry as used.

`\gls`
`\newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}`

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@gls}[2] [] {%
  \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@gls@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentrytext{#2}#3}%
}
```

`\GLStext` behaves like `\gls` except the text is converted to uppercase.

`\GLStext`
`\newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}`
`\glsmfublocker{\GLStext}`

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLStext}[2] [] {%  
  \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLStext@#1#2[#3] {%  
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrytext{#2}#3}}%  
}
```

`\Glstext` behaves like `\gls` except that the first letter of the text is converted to uppercase.

`\Glstext`

```
\newrobustcmd*{\Glstext}{\@gls@hyp@opt\@Glstext}
```

Add sentence case mapping.

```
\glsmfuaddmap{\gls}{\Glstext}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glstext}[2] [] {%  
  \new@ifnextchar[{\@Glstext@{#1}{#2}}{\@Glstext@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@Glstext@#1#2[#3] {%  
  \@gls@field@link{#1}{#2}{\Glsentrytext{#2}#3}%  
}
```

`\glsfirst` behaves like `\gls` except it always uses the value given by the first key and it doesn't mark the entry as used.

`\glsfirst`

```
\newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsfirst}[2] [] {%  
  \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsfirst@#1#2[#3] {%  
  \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%  
}
```

`\Glsfirst` behaves like `\glsfirst` except it displays the first letter in uppercase.

`\Glsfirst`

```
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsfirst}{\Glsfirst}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\Glsfirst}[2] [] {%
  \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@Glsfirst@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
```

`\GLSfirst` behaves like `\Glsfirst` except it displays the text in uppercase.

`\GLSfirst`

```
\newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
\glsmfublocker{\GLSfirst}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSfirst}[2] [] {%
  \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSfirst@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\Glsentryfirst{#2}#3}}%
}
```

`\glsplural` behaves like `\gls` except it always uses the value given by the plural key and it doesn't mark the entry as used.

`\glsplural`

```
\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsplural}[2] [] {%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
```

`\Glsplural` behaves like `\glsplural` except that the first letter is converted to uppercase.

`\Glsplural`

```
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsplural}{\Glsplural}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsplural}[2] []{%  
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@Glsplural@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@glsentryplural{#2}#3}%  
}
```

`\GLSplural` behaves like `\glsplural` except that the text is converted to uppercase.

`\GLSplural`

```
\newrobustcmd*{\@GLSplural}{\@gls@hyp@opt\@GLSplural}  
\glsmfublocker{\@GLSplural}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSplural}[2] []{%  
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSplural@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@glsuppercase{\@glsentryplural{#2}#3}%  
}
```

`\glsfirstplural` behaves like `\gls` except it always uses the value given by the `firstplural` key and it doesn't mark the entry as used.

`\glsfirstplural`

```
\newrobustcmd*{\@glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsfirstplural}[2] []{%  
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsfirstplural@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@glsentryfirstplural{#2}#3}%  
}
```

`\Glsfirstplural` behaves like `\glsfirstplural` except that the first letter is converted to uppercase.

`\Glsfirstplural`

```
\newrobustcmd*{\@Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
```

Add sentence case mapping.

```
\glsmfuaddmap{\@glsfirstplural}{\@Glsfirstplural}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsfirstplural}[2] []{%  
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@Glsfirstplural@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@Glsentryfirstplural{#2}#3}%  
}
```

`\GLSfirstplural` behaves like `\glsfirstplural` except that the link text is converted to uppercase.

`\GLSfirstplural`

```
\newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}  
\glsmfublocker{\GLSfirstplural}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSfirstplural}[2] []{%  
  \new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSfirstplural@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@glsuppercase{\@Glsentryfirstplural{#2}#3}%  
}
```

`\glsname` behaves like `\gls` except it always uses the value given by the name key and it doesn't mark the entry as used.

`\glsname`

```
\newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsname}[2] []{%  
  \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsname@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@Glsentryname{#2}#3}%  
}
```

`\Glsname` behaves like `\glsname` except that the first letter is converted to uppercase.

`\Glsname`

```
\newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsname}{\Glsname}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLsname}[2] [] {%  
  \new@ifnextchar[{\@GLsname@{#1}{#2}}{\@GLsname@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLsname@#1#2[#3] {%  
  \@gls@field@link{#1}{#2}{\@gls@entryname{#2}#3}%  
}
```

`\GLSname` behaves like `\glsname` except that the link text is converted to uppercase.

`\GLSname`

```
\newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}  
\glsmfublocker{\GLSname}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSname}[2] [] {%  
  \new@ifnextchar[{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSname@#1#2[#3] {%  
  \@gls@field@link{#1}{#2}{\@gls@supercase{\@gls@entryname{#2}#3}}%  
}
```

`\glsdesc` behaves like `\gls` except it always uses the value given by the description key and it doesn't mark the entry as used.

`\glsdesc`

```
\newrobustcmd*{\glsdesc}{\@gls@hyp@opt\@glsdesc}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsdesc}[2] [] {%  
  \new@ifnextchar[{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsdesc@#1#2[#3] {%  
  \@gls@field@link{#1}{#2}{\@gls@entrydesc{#2}#3}%  
}
```

`\Glsdesc` behaves like `\glsdesc` except that the first letter is converted to uppercase.

`\Glsdesc`

```
\newrobustcmd*{\Glsdesc}{\@gls@hyp@opt\@Glsdesc}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsdesc}{\Glsdesc}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\GLSdesc}[2] [] {%  
  \new@ifnextchar[{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSdesc@#1#2[#3] {%  
  \@gls@field@link{#1}{#2}{\glsentrydesc{#2}#3}%  
}
```

`\GLSdesc` behaves like `\glsdesc` except that the link text is converted to uppercase.

`\GLSdesc`

```
\newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}  
\glsmfublocker{\GLSdesc}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSdesc}[2] [] {%  
  \new@ifnextchar[{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSdesc@#1#2[#3] {%  
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrydesc{#2}#3}}%  
}
```

`\glsdescplural` behaves like `\gls` except it always uses the value given by the `descriptionplural` key and it doesn't mark the entry as used.

`\glsdescplural`

```
\newrobustcmd*{\glsdescplural}{\@gls@hyp@opt\@glsdescplural}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsdescplural}[2] [] {%  
  \new@ifnextchar[{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsdescplural@#1#2[#3] {%  
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%  
}
```

`\Glsdescplural` behaves like `\glsdescplural` except that the first letter is converted to uppercase.

`\Glsdescplural`

```
\newrobustcmd*{\Glsdescplural}{\@gls@hyp@opt\@Glsdescplural}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsdescplural}{\Glsdescplural}
```


Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\GLSdescplural}[2][\%
  \new@ifnextchar[{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2}[]}]}
```

Read in the final optional argument:

```
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
}
```

`\GLSdescplural` behaves like `\glsdescplural` except that the link text is converted to uppercase.

`\GLSdescplural`

```
\newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\glsmfublocker{\GLSdescplural}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSdescplural}[2][\%
  \new@ifnextchar[{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2}[]}]}
```

Read in the final optional argument:

```
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\Glsentrydescplural{#2}#3}%
}
```

`\glsymbol` behaves like `\gls` except it always uses the value given by the symbol key and it doesn't mark the entry as used.

`\glsymbol`

```
\newrobustcmd*{\glsymbol}{\@gls@hyp@opt\@glsymbol}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsymbol}[2][\%
  \new@ifnextchar[{\@glsymbol@{#1}{#2}}{\@glsymbol@{#1}{#2}[]}]}
```

Read in the final optional argument:

```
\def\@glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrysymbol{#2}#3}%
}
```

`\Glsymbol` behaves like `\glsymbol` except that the first letter is converted to uppercase.

`\Glsymbol`

```
\newrobustcmd*{\Glsymbol}{\@gls@hyp@opt\@Glsymbol}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsymbol}{\Glsymbol}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLssymbol}[2] []{%
  \new@ifnextchar[{\@GLssymbol@{#1}{#2}}{\@GLssymbol@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@glsentrysymbol{#2}#3}%
}
```

`\GLSsymbol` behaves like `\glsymbol` except that the link text is converted to uppercase.

`\GLSsymbol`

```
\newrobustcmd*{\GLSsymbol}{\@gls@hyp@opt\@GLSsymbol}
\glsmfublocker{\GLSsymbol}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSsymbol}[2] []{%
  \new@ifnextchar[{\@GLSsymbol@{#1}{#2}}{\@GLSsymbol@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@glsuppercase{\@glsentrysymbol{#2}#3}%
}
```

`\glsymbolplural` behaves like `\gls` except it always uses the value given by the `symbolplural` key and it doesn't mark the entry as used.

`\glsymbolplural`

```
\newrobustcmd*{\glsymbolplural}{\@gls@hyp@opt\@glsymbolplural}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsymbolplural}[2] []{%
  \new@ifnextchar[{\@glsymbolplural@{#1}{#2}}{\@glsymbolplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@glsentrysymbolplural{#2}#3}%
}
```

`\Glsymbolplural` behaves like `\glsymbolplural` except that the first letter is converted to uppercase.

`\Glsymbolplural`

```
\newrobustcmd*{\Glsymbolplural}{\@gls@hyp@opt\@Glsymbolplural}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsymbolplural}{\Glsymbolplural}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsymbolplural}[2] [] {%  
  \new@ifnextchar[{\@Glsymbolplural@{#1}{#2}}{\@Glsymbolplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@Glsymbolplural@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@glsentrysymbolplural{#2}#3}%  
}
```

`\GLSsymbolplural` behaves like `\glsymbolplural` except that the link text is converted to uppercase.

`\GLSsymbolplural`

```
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}  
\glsmfublocker{\GLSsymbolplural}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSsymbolplural}[2] [] {%  
  \new@ifnextchar[{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSsymbolplural@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@glsuppercase{\@glsentrysymbolplural{#2}#3}}%  
}
```

`\glsuseri` behaves like `\gls` except it always uses the value given by the `user1` key and it doesn't mark the entry as used.

`\glsuseri`

```
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsuseri}[2] [] {%  
  \new@ifnextchar[{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsuseri@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@glsentryuseri{#2}#3}%  
}
```

`\Glsuseri` behaves like `\glsuseri` except that the first letter is converted to uppercase.

`\Glsuseri`

```
\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsuseri}{\Glsuseri}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsuseri}[2] [] {%
  \new@ifnextchar[{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@Glsuseri@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\@Glsentryuseri{#2}#3}%
}
```

`\GLSuseri` behaves like `\glsuseri` except that the link text is converted to uppercase.

`\GLSuseri`

```
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\glsmfublocker{\GLSuseri}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSuseri}[2] [] {%
  \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSuseri@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\@glsuppercase{\@Glsentryuseri{#2}#3}%
}
```

`\glsuserii` behaves like `\gls` except it always uses the value given by the `user2` key and it doesn't mark the entry as used.

`\glsuserii`

```
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsuserii}[2] [] {%
  \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsuserii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\@Glsentryuserii{#2}#3}%
}
```

`\Glsuserii` behaves like `\glsuserii` except that the first letter is converted to uppercase.

`\Glsuserii`

```
\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsuserii}{\Glsuserii}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsuserii}[2] []{%  
  \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@Glsuserii@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@glsentryuserii{#2}#3}%  
}
```

\Glsuserii behaves like \glsuserii except that the link text is converted to uppercase.

\Glsuserii

```
\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}  
\glsmfublocker{\Glsuserii}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSuserii}[2] []{%  
  \new@ifnextchar[{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSuserii@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@glsuppercase{\@glsentryuserii{#2}#3}%  
}
```

\glsuseriii behaves like \gls except it always uses the value given by the user3 key and it doesn't mark the entry as used.

\glsuseriii

```
\newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsuseriii}[2] []{%  
  \new@ifnextchar[{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsuseriii@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@glsentryuseriii{#2}#3}%  
}
```

\Glsuseriii behaves like \glsuseriii except that the first letter is converted to uppercase.

\Glsuseriii

```
\newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsuseriii}{\Glsuseriii}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLsuseriii}[2] [] {%
  \new@ifnextchar[{\@GLsuseriii@{#1}{#2}}{\@GLsuseriii@{#1}{#2} []}]}
```

Read in the final optional argument:

```
\def\@GLsuseriii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
}
```

\GLSuseriii behaves like \glsuseriii except that the link text is converted to uppercase.

\GLSuseriii

```
\newrobustcmd*{\GLSuseriii}{\@gls@hyp@opt\@GLSuseriii}
\glsmfublocker{\GLSuseriii}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSuseriii}[2] [] {%
  \new@ifnextchar[{\@GLSuseriii@{#1}{#2}}{\@GLSuseriii@{#1}{#2} []}]}
```

Read in the final optional argument:

```
\def\@GLSuseriii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\Glsentryuseriii{#2}#3}}%
}
```

\glsuseriv behaves like \gls except it always uses the value given by the user4 key and it doesn't mark the entry as used.

\glsuseriv

```
\newrobustcmd*{\glsuseriv}{\@gls@hyp@opt\@glsuseriv}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsuseriv}[2] [] {%
  \new@ifnextchar[{\@glsuseriv@{#1}{#2}}{\@glsuseriv@{#1}{#2} []}]}
```

Read in the final optional argument:

```
\def\@glsuseriv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryuseriv{#2}#3}%
}
```

\Glsuseriv behaves like \glsuseriv except that the first letter is converted to uppercase.

\Glsuseriv

```
\newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsuseriv}{\Glsuseriv}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsuseriv}[2] []{%
  \new@ifnextchar[{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@Glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@glsentryuseriv{#2}#3}%
}
```

\GLSuseriv behaves like \glsuseriv except that the link text is converted to uppercase.

\GLSuseriv

```
\newrobustcmd*{\GLSuseriv}{\@gls@hyp@opt\@GLSuseriv}
\glsmfublocker{\GLSuseriv}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSuseriv}[2] []{%
  \new@ifnextchar[{\@GLSuseriv@{#1}{#2}}{\@GLSuseriv@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@glsuppercase{\@glsentryuseriv{#2}#3}%
}
```

\glsuserv behaves like \gls except it always uses the value given by the user5 key and it doesn't mark the entry as used.

\glsuserv

```
\newrobustcmd*{\glsuserv}{\@gls@hyp@opt\@glsuserv}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsuserv}[2] []{%
  \new@ifnextchar[{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@glsentryuserv{#2}#3}%
}
```

\Glsuserv behaves like \glsuserv except that the first letter is converted to uppercase.

\Glsuserv

```
\newrobustcmd*{\Glsuserv}{\@gls@hyp@opt\@Glsuserv}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsuserv}{\Glsuserv}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsuserv}[2][ ]{%  
  \new@ifnextchar[{\@Glsuserv@{#1}{#2}}{\@Glsuserv@{#1}{#2}[ ]}}
```

Read in the final optional argument:

```
\def\@Glsuserv@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@Glsentryuserv{#2}#3}%  
}
```

\Glsuserv behaves like \glsuserv except that the link text is converted to uppercase.

\Glsuserv

```
\newrobustcmd*{\Glsuserv}{\@gls@hyp@opt\@Glsuserv}  
\glsmfublocker{\Glsuserv}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsuserv}[2][ ]{%  
  \new@ifnextchar[{\@Glsuserv@{#1}{#2}}{\@Glsuserv@{#1}{#2}[ ]}}
```

Read in the final optional argument:

```
\def\@Glsuserv@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@glsuppercase{\@Glsentryuserv{#2}#3}%  
}
```

\glsuservi behaves like \gls except it always uses the value given by the user6 key and it doesn't mark the entry as used.

\glsuservi

```
\newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsuservi}[2][ ]{%  
  \new@ifnextchar[{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2}[ ]}}
```

Read in the final optional argument:

```
\def\@glsuservi@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\@Glsentryuservi{#2}#3}%  
}
```

\Glsuservi behaves like \glsuservi except that the first letter is converted to uppercase.

\Glsuservi

```
\newrobustcmd*{\Glsuservi}{\@gls@hyp@opt\@Glsuservi}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsuservi}{\Glsuservi}
```


Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsuservi}[2] []{%
  \new@ifnextchar[{\@Glsuservi@{#1}{#2}}{\@Glsuservi@{#1}{#2} []]}
```

Read in the final optional argument:

```
\def\@Glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@glsentryuservi{#2}#3}%
}
```

\GLSuservi behaves like \glsuservi except that the link text is converted to uppercase.

\GLSuservi

```
\newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
\glsmfublocker{\GLSuservi}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSuservi}[2] []{%
  \new@ifnextchar[{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2} []]}
```

Read in the final optional argument:

```
\def\@GLSuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@glsuppercase{\@glsentryuservi{#2}#3}}%
}
```

Now deal with acronym related keys. First the short form:

\acrshort

```
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\@ns@acrshort}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@ns@acrshort}[2] []{%
  \new@ifnextchar[{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2} []}]%
}
```

Read in the final optional argument:

```
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \let\glsifplural\@secondoftwo
    \let\glsifscaps\@firstofthree
    \let\glsinsert\@empty
    \def\glsinserttext{%
      \acronymfont{\@glsentryshort{#2}}#3%
    }%
  }
```

Call `\@gls@link` Note that `\@gls@link` sets `\glstyp`.

```
\@gls@link[#1]{#2}{\csname gls@glstyp @entryfmt\endcsname}%
}%

\glspostlinkhook
}
```

`\Acrshort`

```
\newrobustcmd*{\Acrshort}{\@gls@hyp@opt\ns@Acrshort}
```

Add sentence case mapping.

```
\glsmfuaddmap{\Acrshort}{\Acrshort}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\ns@Acrshort}[2] [] {%
  \new@ifnextchar[{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2} []}]%
}
```

Read in the final optional argument:

```
\def\@Acrshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%

    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsifscaps\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
  }%
}
```

Call `\@gls@link` Note that `\@gls@link` sets `\glstyp`.

```
\@gls@link[#1]{#2}{\csname gls@glstyp @entryfmt\endcsname}%
}%

\glspostlinkhook
}
```

`\ACRshort`

```
\newrobustcmd*{\ACRshort}{\@gls@hyp@opt\ns@ACRshort}
\glsmfublocker{\ACRshort}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\ns@ACRshort}[2] [] {%
  \new@ifnextchar[{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}]%
}
```

Read in the final optional argument:

```
\def\@ACRshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsuppercase{\acronymfont{\glentryshort{#2}}#3}%
    }%
  }
```

Call `\@gl@link` Note that `\@gl@link` sets `\glstype`.

```
\@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
```

```
\glspostlinkhook
}
```

Short plural:

`\acrshortpl`

```
\newrobustcmd*{\acrshortpl}{\@gl@hyp@opt\@ns@acrshortpl}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\ns@acrshortpl}[2] [] {%
  \new@ifnextchar[{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} [] }%
}
```

Read in the final optional argument:

```
\def\@acrshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glentryshortpl{#2}}#3%
    }%
  }
```

Call `\@gl@link` Note that `\@gl@link` sets `\glstype`.

```
\@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
```

```

    \glspostlinkhook
  }

\Acrshortpl
  \newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\ns@Acrshortpl}

Add sentence case mapping.
  \glsmfuaddmap{\Acrshortpl}{\Acrshortpl}
Define the un-starred form. Need to determine if there is a final optional argument
  \newcommand*{\ns@Acrshortpl}[2][{}]{%
    \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2}[]}%
  }
Read in the final optional argument:
  \def\@Acrshortpl#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

      \def\glslabel{#2}%
      \let\glsifplural\@firstoftwo
      \let\glscapscase\@secondofthree
      \let\glsinsert\@empty
      \def\glscustomtext{%
        \acronymfont{\Glsentryshortpl{#2}}#3%
      }%
    }%
  }

Call \@gls@link Note that \@gls@link sets \glsstyle.
  \@gls@link[#1]{#2}{\csname gls@\glsstyle @entryfmt\endcsname}%
  }%

  \glspostlinkhook
}

\ACRshortpl
  \newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\ns@ACRshortpl}
  \glsmfublocker{\ACRshortpl}

Define the un-starred form. Need to determine if there is a final optional argument
  \newcommand*{\ns@ACRshortpl}[2][{}]{%
    \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2}[]}%
  }
Read in the final optional argument:
  \def\@ACRshortpl#1#2[#3]{%
    \glsdoifexists{#2}%
    {%

```

```

\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glsescapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \glsuppercase{\acronymfont{\glsentryshortpl{#2}}#3}%
}%

```

Call `\@gls@link` Note that `\@gls@link` sets `\glstype`.

```

\@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%

```

```

\glspostlinkhook
}

```

`\acrlong`

```

\newrobustcmd*{\acrlong}{\@gls@hyp@opt\@ns@acrlong}

```

Define the un-starred form. Need to determine if there is a final optional argument

```

\newcommand*{\ns@acrlong}[2][{}]{%
  \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2}[]}%
}

```

Read in the final optional argument:

```

\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsescapscase\@firstofthree
    \let\glsinsert\@empty

```

Bug fix v4.02 removed `\acronymfont` from `\glscustomtext` (`\acronymfont` only designed for short form).

```

\def\glscustomtext{%
  \glsentrylong{#2}#3%
}%

```

Call `\@gls@link` Note that `\@gls@link` sets `\glstype`.

```

\@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%

```

```

\glspostlinkhook
}

```

`\Acrlong`

```
\newrobustcmd*{\Acrlong}{\@gls@hyp@opt\ns@Acrlong}
```

Add sentence case mapping.

```
\glsmfuaddmap{\Acrlong}{\Acrlong}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\ns@Acrlong}[2] []{%  
  \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2} []}%  
}
```

Read in the final optional argument:

```
\def\@Acrlong#1#2[#3]{%  
  \glsdoifexists{#2}%  
  {%  
  
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper  
  
    \def\glslabel{#2}%  
    \let\glsifplural\@secondoftwo  
    \let\glscapscase\@secondofthree  
    \let\glsinsert\@empty
```

Bug fix v4.02 removed `\acronymfont` from `\glscustomtext` (`\acronymfont` only designed for short form).

```
\def\glscustomtext{%  
  \Glsentrylong{#2}#3%  
}%
```

Call `\@gls@link`. Note that `\@gls@link` sets `\glstype`.

```
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%  
}%  
  
\glspostlinkhook  
}
```

`\ACRlong`

```
\newrobustcmd*{\ACRlong}{\@gls@hyp@opt\ns@ACRlong}  
\glsmfublocker{\ACRlong}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\ns@ACRlong}[2] []{%  
  \new@ifnextchar[{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2} []}%  
}
```

Read in the final optional argument:

```
\def\@ACRlong#1#2[#3]{%  
  \glsdoifexists{#2}%  
  {%
```

```
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
```

```
\def\glslabel{#2}%
\let\glsifplural\@secondoftwo
\let\glsescapscase\@thirdofthree
\let\glsinsert\@empty
```

Bug fix v4.02 removed `\acronymfont` from `\glscustomtext` (`\acronymfont` only designed for short form).

```
\def\glscustomtext{%
  \glsuppercase{\glsentrylong{#2}#3}%
}%
```

Call `\@gls@link`. Note that `\@gls@link` sets `\glstype`.

```
\@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
```

```
\glspostlinkhook
}
```

Short plural:

`\acrlongpl`

```
\newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\@ns@acrlongpl}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\ns@acrlongpl}[2][ ]{%
  \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2}[ ]}%
}
```

Read in the final optional argument:

```
\def\@acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
```

```
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
```

```
\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glsescapscase\@firstofthree
\let\glsinsert\@empty
```

Bug fix v4.02 removed `\acronymfont` from `\glscustomtext` (`\acronymfont` only designed for short form).

```
\def\glscustomtext{%
  \glsentrylongpl{#2}#3%
}%
```

Call `\@gls@link`. Note that `\@gls@link` sets `\glstype`.

```
\@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
```

```

    \glspostlinkhook
  }

\Acrlongpl
  \newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\ns@Acrlongpl}

Add sentence case mapping.
  \glsmfuaddmap{\Acrlongpl}{\Acrlongpl}
Define the un-starred form. Need to determine if there is a final optional argument
  \newcommand*{\ns@Acrlongpl}[2][ ]{%
    \new@ifnextchar[{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2} [ ]}%
  }
Read in the final optional argument:
  \def\@Acrlongpl#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

      \def\glslabel{#2}%
      \let\glsifplural\@firstoftwo
      \let\glscapscase\@secondofthree
      \let\glsinsert\@empty

Bug fix v4.02 removed \acronymfont from \glscustomtext (\acronymfont
only designed for short form).
      \def\glscustomtext{%
        \Glsentrylongpl{#2}#3%
      }%
Call \@gls@link. Note that \@gls@link sets \glstype.
      \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
    }%

    \glspostlinkhook
  }

\ACRlongpl
  \newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\ns@ACRlongpl}
  \glsmfublocker{\ACRlongpl}

Define the un-starred form. Need to determine if there is a final optional argument
  \newcommand*{\ns@ACRlongpl}[2][ ]{%
    \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2} [ ]}%
  }

```


`\@Gls@entry@field` Generic first letter uppercase version.

```
\@Gls@entry@field{\label}{\field}
```

Check for PDF bookmarks performed by `\@Glsentryfield`.

```
\newrobustcmd*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@#2}%
    \ifdef\@glo@text
    {%
      \expandafter\glsentencecase\expandafter{\@glo@text}%
    }%
    {%
      ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
        entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
        label and the field name}%
    }%
  }%
  {%
    ???%
  }%
}
```

Aside from `\Glsentryname`, the `\Glsentry...` sentence case commands can all follow the same format.

```
\@Glsentryfield{\entry-label}{\field-label}
```

`\@Glsentryfield`

```
\newcommand{\@Glsentryfield}[2]{%
  \glstexorpdfstring{\@Gls@entry@field{#1}{#2}}%
  {\MFUsentencecase{\@Gls@entry@field{#1}{#2}}}%
}
```

`\glstexorpdfstring` Provide a command that will use `\texorpdfstring` if it has been defined. (Remove all the cumbersome tests to determine whether or not that command has been defined.)

```
\ifdef\texorpdfstring
{\newcommand{\glstexorpdfstring}{\texorpdfstring}}
{\newcommand{\glstexorpdfstring}[2]{#1}}
```

Get the entry name (as specified by the name key when the entry was defined). The argument is the label associated with the entry. Note that unless you used `name=false` in the `sanitize` package option you may get unexpected results if the name key contains any commands.

`\glsentryname`

```
\newcommand*{\glsentryname}[1]{\@Gls@entry@field{#1}{name}}
```

The sentence-case equivalents were made robust since they use the non-expandable `\makefirstuc`. These will now expand in the PDF bookmarks to sentence case using `\MFUsentencecase` (which is expandable). Obviously, if any of the fields contain non-expandable content (such as acronym font changing commands) then that will cause a problem (which can be dealt with by using `hyperref`'s `\pdfstringdefDisableCommands`).

```
\Glsentryname
  \newcommand*\Glsentryname[1]{%
    \glstexorpdfstring{\protect\@Gls@entryname{#1}}%
    {\MFUsentencecase{\@gls@entry@field{#1}{name}}}%
  }

  \glsmfuexcl{\Glsentryname}
```

`\@Gls@entryname` This is a workaround in the event that the user defies the warning in the manual about not using `\Glsname` or `\Glsentryname` with acronyms. The check for PDF bookmarks is in `\Glsentryname`. First the default behaviour:

```
\newcommand*\@Gls@entryname[1]{%
  \@Gls@entry@field{#1}{name}%
}
```

`\@Gls@acentryname` Now the behaviour when `\setacronymstyle` is used. The original definition of this command used `mfirstuc` internal commands that are now being deprecated.

```
\newrobustcmd*\@Gls@acentryname[1]{%
  \ifglshaslong{#1}%
  {%
```

Just need to ensure that `\acrnameformat` is expanded, which may cause problems as non-robust semantic commands may expand to content that interferes with sentence-casing.

```
  \protected@edef\@glo@text{\@gls@entry@field{#1}{name}}%
  \expandafter\glsentencecase\expandafter{\@glo@text}%
  }%
  {%
```

Not an acronym

```
  \@Gls@entry@field{#1}{name}%
  }%
}
```

`\@org@Gls@acentryname` Original definition of `\@Gls@acentryname`. This will be removed in a later version.

```
\newcommand*\@org@Gls@acentryname[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@name}%
  }
```

`\@gls@getbody` is defined by `mfirstuc` (which used to be part of `glossaries`).

```

\expandafter\@gls@getbody\@glo@text{}\@nil
\expandafter\ifx\@gls@body\glsentrylong\relax
\expandafter\Glsentrylong\@gls@rest
\else
\expandafter\ifx\@gls@body\glsentryshort\relax
\expandafter\Glsentryshort\@gls@rest
\else
\expandafter\ifx\@gls@body\acronymfont\relax

```

Temporarily make `\glsentryshort` behave like `\Glsentryshort`. (This is on the assumption that the argument of `\acronymfont` is `\glsentryshort` `{\label}`}, as that's the behaviour of the predefined acronym styles.) This is scoped to localise the effect of the assignment.

```

{
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
{
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\expandafter\glsentencecase\expandafter{\@glo@text}%
\fi
\fi
\fi
\fi
}%
{

```

Not an acronym

```

\@Gls@entry@field{#1}{name}%
}%
}

```

Get the entry description (as specified by the `description` when the entry was defined). The argument is the label associated with the entry. Note that unless you used `description=false` in the `sanitize` package option you may get unexpected results if the `description` key contained any commands.

```

\glsentrydesc
\newcommand*\glsentrydesc[1]{\@gls@entry@field{#1}{desc}}

\Glsentrydesc
\newcommand*\Glsentrydesc[1]{\@Gls@entry@field{#1}{desc}}

```

Version 4.54: sentence case mapping removed as `\glsentrydesc` is expandable. This ensures that `\glsentrydesc` can be converted to all-caps if required. However, `\Glsentrydesc` needs to be added as an exclusion. (Similarly for the other `\Glsentry<field>` commands)

```
\glsmfuexcl{\Glsentrydesc}
```

Plural form:

```
\glsentrydescplural
```

```
\newcommand*{\glsentrydescplural}[1]{%
  \@gls@entry@field{#1}{descplural}%
}
```

```
\Glsentrydescplural
```

```
\newcommand*{\Glsentrydescplural}[1]{%
  \@Glsentryfield{#1}{descplural}%
}
```

```
\glsmfuexcl{\Glsentrydescplural}
```

Get the entry text, as specified by the `text` key when the entry was defined. The argument is the label associated with the entry:

```
\glsentrytext
```

```
\newcommand*{\glsentrytext}[1]{\@gls@entry@field{#1}{text}}
```

```
\Glsentrytext
```

```
\newcommand*{\Glsentrytext}[1]{%
  \@Glsentryfield{#1}{text}%
}
```

```
\glsmfuexcl{\Glsentrytext}
```

Get the plural form:

```
\glsentryplural
```

```
\newcommand*{\glsentryplural}[1]{%
  \@gls@entry@field{#1}{plural}%
}
```

```
\Glsentryplural
```

```
\newcommand*{\Glsentryplural}[1]{%
  \@Glsentryfield{#1}{plural}%
}
```

```
\glsmfuexcl{\Glsentryplural}
```

Get the symbol associated with this entry. The argument is the label associated with the entry.

```
\glsentrysymbol
  \newcommand*\glsentrysymbol[1]{%
    \@gls@entry@field{#1}{symbol}%
  }
```

```
\Glsentrysymbol
  \newcommand*\Glsentrysymbol[1]{%
    \@Glsentryfield{#1}{symbol}%
  }
```

```
\glsmfuexcl{\Glsentrysymbol}
```

Plural form:

```
\glsentrysymbolplural
  \newcommand*\glsentrysymbolplural[1]{%
    \@gls@entry@field{#1}{symbolplural}%
  }
```

```
\Glsentrysymbolplural
  \newcommand*\Glsentrysymbolplural[1]{%
    \@Glsentryfield{#1}{symbolplural}%
  }
```

```
\glsmfuexcl{\Glsentrysymbolplural}
```

Get the entry text to be used when the entry is first used in the document (as specified by the first key when the entry was defined).

```
\glsentryfirst
  \newcommand*\glsentryfirst[1]{%
    \@gls@entry@field{#1}{first}%
  }
```

```
\Glsentryfirst
  \newcommand*\Glsentryfirst[1]{%
    \@Glsentryfield{#1}{first}%
  }
```

```
\glsmfuexcl{\Glsentryfirst}
```

Get the plural form (as specified by the firstplural key when the entry was defined).

```
\glsentryfirstplural
  \newcommand*\glsentryfirstplural[1]{%
    \@gls@entry@field{#1}{firstpl}%
  }
```

```

\Glsentryfirstplural
    \newcommand*{\Glsentryfirstplural}[1]{%
        \@Glsentryfield{#1}{firstpl}%
    }

    \glsmfuexcl{\Glsentryfirstplural}

\glscapitalisewords Make it easier to switch from \capitalisewords to \capitalisefmtwords.
    \newcommand{\glscapitalisewords}[1]{%
        \capitalisewords{#1}%
    }

\glsentrytitlecase Now uses sentence case in PDF bookmark.
    \newcommand*{\glsentrytitlecase}[2]{%
        \glstexorpdfstring
            {\@glsentrytitlecase{#1}{#2}}%
            {\MFUsentencecase{\@gls@entry@field{#1}{#2}}}%
    }

\@glsentrytitlecase
    \newrobustcmd*{\@glsentrytitlecase}[2]{%
        \glsdoifexists{#1}%
        {%
            \glsfieldfetch{#1}{#2}{\@gls@value}%
            \expandafter\glscapitalisewords\expandafter{\@gls@value}%
        }%
    }

    Display the glossary type with which this entry is associated (as specified by
    the type key used when the entry was defined). This expands to a label so isn't
    intended to produce text in the document.

\glsentrytype
    \newcommand*{\glsentrytype}[1]{\@gls@entry@field{#1}{type}}

    Display the sort text used for this entry. Note that the sort key is sanitize,
    so unexpected results may occur if the sort key contained commands. This is
    likewise not intended to produce text in the document, other than for debugging
    purposes.

\glsentrysort
    \newcommand*{\glsentrysort}[1]{%
        \@gls@entry@field{#1}{sort}%
    }

\glsentryparent Expands to the label of the entry's parent. This expands to a label so isn't
    intended to produce text in the document.
    \newcommand*{\glsentryparent}[1]{%
        \@gls@entry@field{#1}{parent}%
    }

```

`\glsentryuseri` Get the first user key (as specified by the `user1` when the entry was defined). The argument is the label associated with the entry.

```

\newcommand*\glsentryuseri[1]{%
  \@gls@entry@field{#1}{useri}%
}

```

`\Glsentryuseri`

```

\newcommand*\Glsentryuseri[1]{%
  \@Glsentryfield{#1}{useri}%
}

\glsmfuexcl{\Glsentryuseri}

```

`\glsentryuserii` Get the second user key (as specified by the `user2` when the entry was defined). The argument is the label associated with the entry.

```

\newcommand*\glsentryuserii[1]{%
  \@gls@entry@field{#1}{userii}%
}

```

`\Glsentryuserii`

```

\newcommand*\Glsentryuserii[1]{%
  \@Glsentryfield{#1}{userii}%
}

\glsmfuexcl{\Glsentryuserii}

```

`\glsentryuseriii` Get the third user key (as specified by the `user3` when the entry was defined). The argument is the label associated with the entry.

```

\newcommand*\glsentryuseriii[1]{%
  \@gls@entry@field{#1}{useriii}%
}

```

`\Glsentryuseriii`

```

\newcommand*\Glsentryuseriii[1]{%
  \@Glsentryfield{#1}{useriii}%
}

Add sentence case mapping.
\glsmfuexcl{\Glsentryuseriii}

```

`\glsentryuseriv` Get the fourth user key (as specified by the `user4` when the entry was defined). The argument is the label associated with the entry.

```

\newcommand*\glsentryuseriv[1]{%
  \@gls@entry@field{#1}{useriv}%
}

```



```

\Glsentryuseriv
    \newcommand*{\Glsentryuseriv}[1]{%
        \@Glsentryfield{#1}{useriv}%
    }

    \glsmfuexcl{\Glsentryuseriv}

\glsentryuseriv Get the fifth user key (as specified by the user5 when the entry was defined).
The argument is the label associated with the entry.
    \newcommand*{\glsentryuseriv}[1]{%
        \@gls@entry@field{#1}{useriv}%
    }

\Glsentryuseriv
    \newcommand*{\Glsentryuseriv}[1]{%
        \@Glsentryfield{#1}{useriv}%
    }

    \glsmfuexcl{\Glsentryuseriv}

\glsentryuseriv Get the sixth user key (as specified by the user6 when the entry was defined).
The argument is the label associated with the entry.
    \newcommand*{\glsentryuseriv}[1]{%
        \@gls@entry@field{#1}{useriv}%
    }

\Glsentryuseriv
    \newcommand*{\Glsentryuseriv}[1]{%
        \@Glsentryfield{#1}{useriv}%
    }

    \glsmfuexcl{\Glsentryuseriv}

\glsentryshort Get the short key (as specified by the short the entry was defined). The argument
is the label associated with the entry.
    \newcommand*{\glsentryshort}[1]{\@gls@entry@field{#1}{short}}

\Glsentryshort
    \newcommand*{\Glsentryshort}[1]{%
        \@Glsentryfield{#1}{short}%
    }

    \glsmfuexcl{\Glsentryshort}

\glsentryshortpl Get the short plural key (as specified by the shortplural the entry was defined).
The argument is the label associated with the entry.
    \newcommand*{\glsentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}

```

```

\Glsentryshortpl
    \newcommand*\Glsentryshortpl}[1]{%
        \@Glsentryfield{#1}{shortpl}%
    }

    \glsmfuexcl{\Glsentryshortpl}

\glsentrylong Get the long key (as specified by the long the entry was defined). The argument
is the label associated with the entry.
    \newcommand*\glsentrylong}[1]{\@gls@entry@field{#1}{long}}

\Glsentrylong
    \newcommand*\Glsentrylong}[1]{%
        \@Glsentryfield{#1}{long}%
    }

    \glsmfuexcl{\Glsentrylong}

\glsentrylongpl Get the long plural key (as specified by the longplural the entry was defined).
The argument is the label associated with the entry.
    \newcommand*\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}

\Glsentrylongpl
    \newcommand*\Glsentrylongpl}[1]{%
        \@Glsentryfield{#1}{longpl}%
    }

    \glsmfuexcl{\Glsentrylongpl}
    Short cut macros to access full form:

\glsentryfull This definition shouldn't actually be used. This command will be redefined by
the acronym style.
    \newcommand*\glsentryfull}[1]{%
        \glsentrylong{#1} (\glsentryshort{#1})%
    }

\Glsentryfull This definition shouldn't actually be used. This command will be redefined by
the acronym style.
    \newcommand*\Glsentryfull}[1]{%
        \Glsentrylong{#1} (\glsentryshort{#1})%
    }

    \glsmfuexcl{\Glsentryfull}

\glsentryfullpl This definition shouldn't actually be used. This command will be redefined by
the acronym style.
    \newcommand*\glsentryfullpl}[1]{%
        \acrffullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
    }

```

`\Glsentryfullpl` This definition shouldn't actually be used. This command will be redefined by the acronym style.

```
\newcommand*\Glsentryfullpl}[1]{%
  \Glsentrylongpl{#1} (\Glsentryshortpl{#1})%
}
%\changes{4.54}{2024-04-03}{change map to exclusion}

\glsmfuexcl{\Glsentryfullpl}
```

`\glsentrynumberlist` Displays the number list as is.

```
\newcommand*\glsentrynumberlist}[1]{%
  \glsdoifexists{#1}%
  {%
    \@gls@entry@field{#1}{numberlist}%
  }%
}
```

`\glsdisplaynumberlist` Formats the number list for the given entry label. Doesn't work with hyperref.

```
\@ifpackageloaded{hyperref} {%
  \newcommand*\glsdisplaynumberlist}[1]{%
    \GlossariesWarning
    {%
      \string\glsdisplaynumberlist\space
      doesn't work with hyperref.^^JUsing
      \string\glsentrynumberlist\space instead%
    }%
    \glsentrynumberlist{#1}%
  }%
}%
{%
  \newcommand*\glsdisplaynumberlist}[1]{%
    \glsdoifexists{#1}%
    {%
      \bgroup

      \protected@edef\@glo@label{\glsdetoklabel{#1}}%
      \let\@org@glnumberformat\glnumberformat
      \def\glnumberformat##1{##1}%
      \protected@edef\the@numberlist{%
        \csname glo@\@glo@label @numberlist\endcsname}%
      \def\@gls@numlist@sep{}%
      \def\@gls@numlist@nextsep{}%
      \def\@gls@numlist@lastsep{}%
      \def\@gls@thislist{}%
      \def\@gls@donext@def{}%
      \renewcommand\do[1]{%
        \protected@edef\@gls@thislist{%
          \@gls@thislist
          \noexpand\@gls@numlist@sep
          ##1%
        }%
      }%
    }%
  }%
}
```

```

    }%
    \let\@gls@numlist@sep\@gls@numlist@nextsep
    \def\@gls@numlist@nextsep{\glsnumlistsep}%
    \@gls@donext@def
    \def\@gls@donext@def{%
      \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
    }%
  }%
  \expandafter \glsnumlistparser \expandafter{\the@numberlist}%
  \let\@gls@numlist@sep\@gls@numlist@lastsep
  \@gls@thislist
\egroup
}%
}
}

```

`\glsnumlistsep`

```
\newcommand*{\glsnumlistsep}{, }
```

`\glsnumlistlastsep`

```
\newcommand*{\glsnumlistlastsep}{ \& }
```

`\gls hyperlink` Provide a hyperlink to a glossary entry without adding information to the glossary file. The entry needs to be added using a command like `\gls link` or `\gls add` to ensure that the target is defined. The first (optional) argument specifies the link text. The entry name is used by default. The second argument is the entry label.

```

\newcommand*{\gls hyperlink}[2][\glsentrytext{\@glo@label}]{%
  \def\@glo@label{#2}%
  \@gls link{\@glo link prefix\@glsdetoklabel{#2}}{#1}}

```

2.12 Adding an entry to the glossary without generating text

The following keys are provided for `\gls add` and `\gls add all`:

```
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
```

```
\define@key{glossadd}{format}{\def\@gls number format{#1}}
```

This key is only used by `\gls add all`:

```
\define@key{glossadd}{types}{\def\@glo@type{#1}}
```

```
\gls add[options]{label}
```

Add a term to the glossary without generating any link text. The optional argument indicates which counter to use, and how to format it (using a key-value list) the second argument is the entry label. Note that *options* only has two keys: counter and format (the types key will be ignored).

`\glsadd`

```
\newrobustcmd*{\glsadd}[2] [] {%
```

Need to move to horizontal mode if not already in it, but only if not in preamble.

```
\@gls@adjustmode  
\glsdoifexists{#2}%  
{%
```

```
\def\@glsnumberformat{glsnumberformat}%
```

```
\protected@edef\@gls@counter{\csname glo@\glsdetoklabel{#2}@counter\endcsname}%  
\setkeys{glossadd}{#1}%
```

Store the entry's counter in `\theglsentrycounter`

```
\@gls@saveentrycounter
```

Define sort key if necessary:

```
\@gls@setsort{#2}%
```

This should use `\@@do@wrglossary` rather than `\@do@wrglossary` since the whole point of `\glsadd` is to add a line to the glossary.

```
\@@do@wrglossary{#2}%  
}%  
}
```

Identify `\glsadd` as an exclusion, but this will only work if the optional argument is omitted. If the optional argument is required, the `\glsadd` command with arguments will need to be encapsulated with `\MFUskipunc`. See the `mfirstuc v2.08+` manual for further details.

```
\glsmfuexcl{\glsadd}
```

`\@gls@adjustmode`

```
\newcommand*{\@gls@adjustmode}{}  
\AtBeginDocument{\renewcommand*{\@gls@adjustmode}{\ifvmode\mbox{} \fi}}
```

The following aren't added as exclusions, since there's no reason for them to occur at the start of content that may require a case-change (such as at the start of a description).

```
\glsaddall[option list]
```

Add all terms defined for the listed glossaries (without displaying any text). If `types` key is omitted, apply to all glossary types.

`\glsaddall`

```
\newrobustcmd*{\glsaddall}[1] [] {%  
\protected@edef\@glo@type{\@glo@types}%  
\setkeys{glossadd}{#1}%  
\forallglsentries[\@glo@type]{\@glo@entry}{%  
\glsadd[#1]{\@glo@entry}%  
}%  
}
```

```
\glsaddallunused[<glossary types>]
```

`\glsaddallunused`

Add all used terms defined for the listed glossaries (without displaying any text). If optional argument is omitted, apply to all glossary types. This should typically go at the end of the document.

```
\newrobustcmd*{\glsaddallunused}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glo@entry}%
  {%
    \ifglsused{\@glo@entry}{\glsadd[format=glsignore]{\@glo@entry}}%
  }%
}
```

`\glsignore`

```
\newcommand*{\glsignore}[1]{}
```

2.13 Creating associated files

The `\writeist` command creates the associated customized `.ist` `makeindex` style file. While defining this command, some characters have their catcodes temporarily changed to ensure they get written to the `.ist` file correctly. The `makeindex` actual character (usually `@`) is redefined to be a `?`, to allow internal commands to be written to the glossary file output file.

The special characters are stored in `\@gls@actualchar`, `\@gls@encapchar`, `\@gls@levelchar` and `\@gls@quotechar` to make them easier to use later, but don't change these values, because the characters are encoded in the command definitions that are used to escape the special characters (which means that the user no longer needs to worry about `makeindex` special characters).

The symbols and numbers label for group headings are hardwired into the `.ist` file as `glsymbols` and `glsnumbers`, the group titles can be translated (so that `\glsymbolsgroupname` replaces `glsymbols` and `\glsnumbersgroupname` replaces `glsnumbers`) using the command `\glsgetgrouptitle` which is defined in `glossary-hypernav`. This is done to prevent any problem characters in `\glsymbolsgroupname` and `\glsnumbersgroupname` from breaking hyperlinks.

`\glsopenbrace` Define `\glsopenbrace` to make it easier to write an opening brace to a file.

```
\edef\glsopenbrace{\expandafter@gobble\string\{}
```

`\glsclosebrace` Define `\glsclosebrace` to make it easier to write an opening brace to a file.

```
\edef\glsclosebrace{\expandafter@gobble\string\}}
```

`\glsbackslash` Define `\glsbackslash` to make it easier to write a backslash to a file.

```
\edef\glsbackslash{\expandafter@gobble\string\}}
```

`\glsquote` Define command that makes it easier to write quote marks to a file in the event that the double quote character has been made active.

```
\edef\glsquote#1{\string"#1\string"}
```

`\glspercentchar` Define `\glspercentchar` to make it easier to write a percent character to a file.

```
\edef\glspercentchar{\expandafter\@gobble\string\%}
```

`\glstildechar` Define `\glstildechar` to make it easier to write a tilde character to a file.

```
\edef\glstildechar{\string~}
```

`\@glsfirstletter` Define the first letter to come after the digits 0, . . . ,9. Only required for xindy.

```
\ifglsxindy
  \newcommand*{\@glsfirstletter}{A}
\fi
```

`\GlsSetXdyFirstLetterAfterDigits` Sets the first letter to come after the digits 0, . . . ,9. The starred version sanitizes.

```
\newcommand*{\GlsSetXdyFirstLetterAfterDigits}{%
  \ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
\ifglsxindy
  \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*{\@glsfirstletter}{#1}}
  \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*{\@glsfirstletter}{#1}%
    \@onelevel@sanitize\@glsfirstletter
  }
\else
  \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \glsnoxywarning\GlsSetXdyFirstLetterAfterDigits}
  \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%
    \GlsSetXdyFirstLetterAfterDigits
  }
\fi
```

`\@xdynumbergrouporder` Specifies the order of the number group.

```
\ifglsxindy
  \newcommand*{\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
\fi
```

`\GlsSetXdyNumberGroupOrder` Sets the relative location of the number group. The starred version sanitizes.

```
\newcommand*{\GlsSetXdyNumberGroupOrder}[1]{%
  \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
}
\ifglsxindy
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
  }
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
    \@onelevel@sanitize\@xdynumbergrouporder
  }
\else
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \glsnoxywarning\GlsSetXdyNumberGroupOrder}
\fi
```

```

        \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
        \@GlsSetXdyNumberGroupOrder}
    \fi

\@glsminrange Define the minimum number of successive location references to merge into a
range.
    \newcommand*{\@glsminrange}{2}

\GlsSetXdyMinRangeLength Set the minimum range length. The value must either be none or a positive
integer. The glossaries package doesn't check if the argument is valid, that is
left to xindy.
    \ifglxindy
        \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
        \renewcommand*{\@glsminrange}{#1}}
    \else
        \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
        \glsnxindywarning\GlsSetXdyMinRangeLength}
    \fi

\writeist
    \ifglxindy
Code to use if xindy is required.
        \def\writeist{%
Define write register if not already defined
            \ifundef{\glswrite}{\newwrite\glswrite}{}%
Update attributes list
            \@glscaddpredefinedattributes
Open the file.
            \openout\glswrite=\istfilename
Write header comment at the start of the file
            \write\glswrite{;; xindy style file created by the glossaries
            package}%
            \write\glswrite{;; for document '\jobname' on
            \the\year-\the\month-\the\day}%
Specify the required styles
            \write\glswrite{^^J; required styles^^J}
            \for\@xdystyle=\@xdyrequiredstyles\do{%
                \ifx\@xdystyle\@empty
                \else
                    \protected@write\glswrite}{(require
                    \string"\@xdystyle.xdy\string")}%
                \fi
            }%
    \fi

```


List the allowed attributes (possible values used by the format key)

```
\write\glswrite{^^J%
; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
```

Define any additional alphabets

```
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
```

Define location classes.

```
\write\glswrite{^^J; location class definitions^^J}%
```

As from version 3.0, locations are now specified as $\langle Hprefix \rangle \langle number \rangle$, so need to add all possible combinations of location types.

```
\@for\@gls@classI:=\@gls@xdy@locationlist\do{%
```

Case where $\langle Hprefix \rangle$ is empty:

```
\protected@write\glswrite{}{(define-location-class
\string"\@gls@classI\string"^^J\space\space\space
(
:sep "{"
\csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
:sep "}"
)
^^J\space\space\space
:min-range-length \@glsminrange^^J%
)
}%
```

Nested iteration over all classes:

```
{%
\@for\@gls@classII:=\@gls@xdy@locationlist\do{%
\protected@write\glswrite{}{(define-location-class
\string"\@gls@classII-\@gls@classI\string"
^^J\space\space\space
(
:sep "{"
\csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
:sep "}"
\csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
:sep "}"
)
^^J\space\space\space
:min-range-length \@glsminrange^^J%
)
}%
}%
}%
}%
```

User defined location classes (needs checking for new location format).

```
\write\glswrite{^^J; user defined location classes}%
```

```
\write\glswrite{\@xdyuserlocationdefs}%
```

Cross-reference class. (The unverified option is used as the cross-references are supplied using the list of labels along with the optional argument for `\glsseeformat` which xindy won't recognise.)

```
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
:unverified )}%
```

Define how cross-references should be displayed. This adds an empty set of braces after the cross-referencing information allowing for the final argument of `\glsseeformat` which gets ignored. (When using `makeindex` this final argument contains the location information which is not required.)

```
\write\glswrite{(markup-crossref-list
: class \string"see\string"^^J\space\space\space
: open \string"\string\glsseeformat\string"
: close \string"{}\string")}%
```

Provide hook to write extra material here (used by `glossaries-extra` to define a `seealso` class).

```
\@xdycrossrefhook
```

List the order to sort the classes.

```
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
(\@xdylocationclassorder))}%
```

Specify what to write to the start and end of the glossary file.

```
\write\glswrite{^^J; define the glossary markup^^J}%

\write\glswrite{(markup-index^^J\space\space\space
: open \string"\string
\glossarysection[\string\glossarytoctitle]{\string
\glossarytitle}\string\glossarypreamble}%
```

Add all the xindy-only macro definitions (needed to prevent errors in the event that the user changes from xindy to `makeindex`)

```
\@for\@this@ctr:=\@xdycounters\do{%
{%
\@for\@this@attr:=\@xdyattributelist\do{%
\protected@write\glswrite{ }\string\providecommand*%
\expandafter\string
\csname glsX\@this@ctr X\@this@attr\endcsname[2]%
{%
\string\setentrycounter
[\expandafter\@gobble\string\#1]{\@this@ctr}%
\expandafter\string
\csname\@this@attr\endcsname
{\expandafter\@gobble\string\#2}%
}%
}%
}%
```

```
}%
}%
```

Add the end part of the open tag and the rest of the markup-index information:

```
\write\glswrite{%
  \string\begin
  {theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
  \space\space:close \string"\glpercentchar\glstildechar n\string
  \end{theglossary}\string\glossarypostamble
  \glstildechar n\string" ^^J\space\space\space
  :tree)}%
```

Specify what to put between letter groups

```
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
```

Specify what to put between entries

```
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \glstildechar n\string")}%
```

Specify how to format entries

```
\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
```

Specify how to separate location numbers

```
\write\glswrite{(markup-locref-list
  :sep \string"\string\delimN\space\string")}%
```

Specify how to indicate location ranges

```
\write\glswrite{(markup-range
  :sep \string"\string\delimR\space\string")}%
```

Specify 2-page and 3-page suffixes, if defined. First, the values must be sanitized to write them explicitly.

```
\@onelevel@sanitize\gls@suffiF
\@onelevel@sanitize\gls@suffiFF

\ifx\gls@suffiF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffiF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffiFF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffiFF" :length 2 :ignore-end)}%
\fi
```

Specify how to format locations.

```
\write\glswrite{^^J; define format to use for locations^^J}%  
\write\glswrite{\@xdylocref}%
```

Specify how to separate letter groups.

```
\write\glswrite{^^J; define letter group list format^^J}%  
\write\glswrite{(markup-letter-group-list  
:sep \string"\string\glsgroupskip\glstildechar n\string")}%
```

Define letter group headings.

```
\write\glswrite{^^J; letter group headings^^J}%  
\write\glswrite{(markup-letter-group  
:open-head \string"\string\glsgroupheading  
\glsopenbrace\string"^^J\space\space\space  
:close-head \string"\glsclosebrace\string")}%
```

Define additional letter groups.

```
\write\glswrite{^^J; additional letter groups^^J}%  
\write\glswrite{\@xdylettergroups}%
```

Define additional sort rules

```
\write\glswrite{^^J; additional sort rules^^J}  
\write\glswrite{\@xdysortrules}%
```

Hook for any additional information:

```
\@gls@writeisthook
```

Close the style file

```
\closeout\glswrite
```

Suppress any further calls.

```
\let\writeist\relax  
\let\noist\@gls@noist@toolate  
}
```

\else

Code to use if makeindex is required.

```
\edef\@gls@actualchar{\string?}  
\edef\@gls@encapchar{\string|}  
\edef\@gls@levelchar{\string!}  
\edef\@gls@quotechar{\string"}%  
\let\GlsSetQuote\gls@nosetquote  
\def\writeist{\relax  
\ifundef{\glswrite}{\newwrite\glswrite}{\relax  
\openout\glswrite=\istfilename  
\write\glswrite{\glspercentchar\space makeindex style file  
created by the glossaries package}  
\write\glswrite{\glspercentchar\space for document  
'\jobname' on \the\year-\the\month-\the\day}  
\write\glswrite{actual '\@gls@actualchar'}  
\write\glswrite{encap '\@gls@encapchar'}  
\write\glswrite{level '\@gls@levelchar'}  
\write\glswrite{quote '\@gls@quotechar'}}
```

```

\write\glswrite{keyword \string\string\glossaryentry\string"}
\write\glswrite{preamble \string\string\glossarysection[\string
  \glossarytoctitle]{\string\glossarytitle}\string
  \glossarypreamble\string\n\string\begin{theglossary}\string
  \glossaryheader\string\n\string"}
\write\glswrite{postamble \string\string%\string\n\string
  \end{theglossary}\string\glossarypostamble\string\n
  \string"}
\write\glswrite{group_skip \string\string\glsgroupskip\string\n
  \string"}
\write\glswrite{item_0 \string\string%\string\n\string"}
\write\glswrite{item_1 \string\string%\string\n\string"}
\write\glswrite{item_2 \string\string%\string\n\string"}
\write\glswrite{item_01 \string\string%\string\n\string"}
\write\glswrite{item_x1
  \string\string\relax \string\glsresetentrylist\string\n
  \string"}
\write\glswrite{item_12 \string\string%\string\n\string"}
\write\glswrite{item_x2
  \string\string\relax \string\glsresetentrylist\string\n
  \string"}

\write\glswrite{delim_0 \string\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_2 \string\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_t \string\string\}\string}\string"}
\write\glswrite{delim_n \string\string\delimN \string"}
\write\glswrite{delim_r \string\string\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
  \string\string\}\string\relax
  \string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string\glsymbols\string"}
\write\glswrite{numhead_positive \string\glsnumbers\string"}
\write\glswrite{page_compositor \string\glscompositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{suffix_2p \string\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{suffix_3p \string\gls@suffixFF\string"}
\fi

```

Hook for any additional information:

```
\@gls@writeisthook
```

Close the file and disable \writeist.

```
\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}
\fi
```

\GlsSetWriteIstHook Allow user to append information to the style file.

```
\newcommand*\GlsSetWriteIstHook[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook
```

\@gls@writeisthook

```
\newcommand*\@gls@writeisthook{}
```

\GlsSetQuote Allow user to set the makeindex quote character. This is primarily for ngerman users who want to use makeindex's -g option.

```
\ifglsxindy
\newcommand*\GlsSetQuote[1]{\glsnomakeindexwarning\GlsSetQuote}
\newcommand*\gls@nosetquote[1]{\glsnomakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote[1]{\edef\@gls@quotechar{\string#1}}%
```

If German is in use, set the extra makeindex option so makeglossaries can pick it up.

```
\@ifpackageloaded{tracklang}%
{%
\IfTrackedLanguage{german}%
{%
\def\@gls@extramakeindexopts{-g}%
}%
}%
}
```

Need to redefine \@gls@checkquote

```
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkquote####1#1####2#1####3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={####1}%
\noexpand\ifx\noexpand\null####2\noexpand\null
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
```

```

\noexpand\@gls@quotechar\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###2#1#1\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###2#1###3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
\noexpand\def\noexpand\@gls@checkedmkidx{%}
\noexpand\expandafter\noexpand\@gls@checkquote###1\noexpand\@nil
#1#1\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%}
\noexpand\expandafter\noexpand\@gls@checkescquote###1\noexpand\@nil
\expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%}
\noexpand\expandafter\noexpand\@gls@checkescactual###1\noexpand\@nil
\noexpand\?\noexpand\?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%}
\noexpand\expandafter\noexpand\@gls@checkactual###1\noexpand\@nil
\noexpand?\noexpand?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%}
\noexpand\expandafter\noexpand\@gls@checkbar###1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%}

```

```

\noexpand\expandafter\noexpand\@gls@checkeschar####1\noexpand\@nil
\noexpand\|\noexpand\|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checklevel####1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%

\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkescllevel####1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%

\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesc####1\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkesquote####1%
\expandonce{\csname#1\endcsname}####2\expandonce{\csname#1\endcsname}%
####3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={####1}%
\noexpand\ifx\noexpand\null####2\noexpand\null
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkesquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkesquote{%
\noexpand\@gls@checkesquote####3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string
\expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%

```



```

\newcommand{\@gls@docheckquotedef}
{
\newcommand*{\@gls@nosetquote}[1]{\PackageError{glossaries}%
{\string\GlsSetQuote\space not permitted here}%
{Move \string\GlsSetQuote\space earlier in the preamble, as
soon as possible after glossaries.sty has been loaded}}
\fi
}

```

`\@gls@extramakeindexopts`

```
\newcommand*{\@gls@extramakeindexopts}[1]{}
```

The command `\noist` will suppress the creation of the `.ist` file. Obviously you need to use this command before `\writeist` to have any effect.

`\noist`

```

\newcommand{\noist}{%
Update attributes list
\@gls@addpredefinedattributes
\let\writeist\relax
}

```

`\@gls@noist@toolate`

```

\newcommand{\@gls@noist@toolate}{%
\GlossariesWarning{too late to use \string\noist. The style file
has already been written}%
}

```

`\@makeglossary` is an internal command that takes an argument indicating the glossary type. This command will create the glossary file required by `makeindex` for the given glossary type, using the extension supplied by the `<out-ext>` parameter used in `\newglossary` (and it will also activate the `\glossary` command, and create the customized `.ist` `makeindex` style file).

Note that you can't use `\@makeglossary` for only some of the defined glossaries (with just the base `glossaries` package). You either need to have a `\makeglossaries` for all glossaries or none (otherwise you will end up with a

situation where T_EX is trying to write to a non-existent file). The relevant glossary must be defined prior to using `\makeglossary`. `glossaries-extra` allows for a hybrid approach.

`\makeglossary` Unstarred form of `\ifglossaryexists` is used as `\makeglossary` can't be used with an ignored glossary.

```
\newcommand*\makeglossary}[1]{%
  \ifglossaryexists{#1}%
  {%
```

Only create a new write if `savewrites=false` otherwise create a token to collect the information.

```
  \ifglssavewrites
    \expandafter\newtoks\csname glo@#1@filetok\endcsname
  \else
    \expandafter\newwrite\csname glo@#1@file\endcsname
    \expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
  \fi
  \@gls@renewglossary
  \writeist
}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘#1’ not defined}%
  {New glossaries must be defined before using \string\makeglossaries}%
}%
}
```

`\glsopenfile` Open write file associated with the given glossary.

```
\newcommand*\glsopenfile}[2]{%
  \immediate\openout#1=\jobname.\csname @glstype@#2@out\endcsname
  \PackageInfo{glossaries}{Writing glossary file
    \jobname.\csname @glstype@#2@out\endcsname}%
}
```

`\@closegls`

```
\newcommand*\@closegls}[1]{%
  \closeout\csname glo@#1@file\endcsname
}
```

`\gls@automake` Unstarred form of `\ifglossaryexists` is used as `\gls@automake` can't be used with an ignored glossary.

```
\ifglsxindy
\newcommand*\gls@automake}[1]{%
  \ifglossaryexists{#1}
  {%
    \@closegls{#1}%
    \ifdefstring{\glsorder}{letter}%
    {\def\@gls@order{-M ord/letorder }}%
  }
```

```

        {\let\@gls@order\@empty}%
\gls@xdy@set@lang{#1}%

\ifdefvoid\gls@codepage
{\let\@gls@codepage\@empty}%
{\def\@gls@codepage{-C \gls@codepage\space}}%
\@gls@run@unrestricted@shell{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \@gls@run@output@file{\gls@istfilebase}
\@gls@codepage
-t \@gls@run@output@file{\jobname.\csuse{@glotype@#1@log}}
-o \@gls@run@output@file{\jobname.\csuse{@glotype@#1@in}}
\@gls@run@output@file{\jobname.\csuse{@glotype@#1@out}}}%
}%
}%
{%
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\else
\newcommand*{\@gls@automake}[1]{%
\ifglossaryexists{#1}
{%
\@closegls{#1}%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-l }}%
{\let\@gls@order\@empty}%
\@gls@runshell{makeindex \@gls@order
-s \@gls@run@output@file{\istfilename}
-t \@gls@run@output@file{\jobname.\csuse{@glotype@#1@log}}
-o \@gls@run@output@file{\jobname.\csuse{@glotype@#1@in}}
\@gls@run@output@file{\jobname.\csuse{@glotype@#1@out}}}%
}%
}%
{%
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\fi

```

`\@gls@automake@immediate` Unstarred form of `\ifglossaryexists` is used as `\@gls@automake@immediate` can't be used with an ignored glossary. Note that “out” means the output file from this package's point of view. It will be the input file from the indexing application's point of view.

```

\ifglsxindy
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%

```

```

\IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
{%
  \ifdefstring{\glsorder}{letter}%
  {\def\@gls@order{-M ord/letorder }}%
  {\let\@gls@order\@empty}%
Set \@gls@langmod:
  \gls@xdy@set@lang{#1}%

  \ifdefvoid\gls@codepage
  {\let\@@gls@codepage\@empty}%
  {\def\@@gls@codepage{-C \gls@codepage\space}}%
  \@gls@run@unrestricted@shell{xindy
  -I xindy
  \@gls@order
  -L \@gls@langmod\space
  -M \@gls@run@output@file{\gls@istfilebase}
  \@gls@codepage
  -t \@gls@run@output@file{\jobname.\csuse{@glotype@#1@log}}
  -o \@gls@run@output@file{\jobname.\csuse{@glotype@#1@in}}
  \@gls@run@output@file{\jobname.\csuse{@glotype@#1@out}}}%
  }%
  }%
  {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
  doesn't exist. Rerun may be required}}%
  }%
  {\%
  \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
  }%
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
  \ifglossaryexists{#1}
  {%
  \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
  {%
  \ifdefstring{\glsorder}{letter}%
  {\def\@gls@order{-l }}%
  {\let\@gls@order\@empty}%
  \@gls@runshell{makeindex \@gls@order
  -s \@gls@run@output@file{\istfilename}
  -t \@gls@run@output@file{\jobname.\csuse{@glotype@#1@log}}
  -o \@gls@run@output@file{\jobname.\csuse{@glotype@#1@in}}
  \@gls@run@output@file{\jobname.\csuse{@glotype@#1@out}}}%
  }%
  }%
  {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
  doesn't exist. Rerun may be required}}%
  }%
  {\%
  \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
  }%
}

```

```

    }%
  }
\fi

```

`\warn@nomakeglossaries` Issue warning that `\makeglossaries` hasn't been used.

```
\newcommand*{\warn@nomakeglossaries}{}

```

Only use this if warning if `\printglossary` has been used without `\makeglossaries`

```
\newcommand*{\warn@nomakeglossaries}{\warn@nomakeglossaries}

```

`\@gls@automake@types` The list of glossaries to build if automake is on. Provided for glossaries-extra's hybrid option.

```
\newcommand{\@gls@automake@types}{\@glo@types}

```

`\@gls@@automake@immediate`

```

\newcommand{\@gls@@automake@immediate}{%
\ifnum\gls@automake@nr=2\relax
  \protected@edef\gls@tmp{\@gls@automake@types}%
  \@for\@gls@type:=\gls@tmp\do{%
    \ifdefempty{\@gls@type}{}%
      {\@gls@automake@immediate{\@gls@type}}%
    }%
  \glsautomakefalse
  \renewcommand*{\@gls@doautomake}{}%
\else
  \ifdefempty\gls@automake@makegloss{}%
    {%
      \IfFileExists{\jobname.aux}%
        {\@gls@run@unrestricted@shell{\@gls@automake@makegloss\space \jobname}}{%}%
      \glsautomakefalse
      \renewcommand*{\@gls@doautomake}{}%
    }%
  \fi
}

```

`\makeglossaries` will use `\@makeglossary` for each glossary type that has been defined. New glossaries need to be defined before using `\makeglossary`, so have `\makeglossaries` redefine `\newglossary` to prevent it being used afterwards.

`\makeglossaries`

```

\newcommand*{\makeglossaries}{%
\ifglxindy
  \def\glsindexingsetting{xindy}%
\else
  \def\glsindexingsetting{makeindex}%
\fi
\@domakeglossaries
}%

```

If automake=immediate setting is on, use the shell escape now.

```
\@gls@automake@immediate
```

Define the write used for style file also used for all other output files if savewrites=true.

```
\ifundef{\glswrite}{\newwrite\glswrite}{}%
```

If the user removes the glossary package from their document, ensure the next run doesn't throw a load of undefined control sequence errors when the aux file is parsed.

```
\protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{}}
\protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{}}
```

If \@gls@extramakeindexopts has been defined, write it:

```
\ifundef\@gls@extramakeindexopts
{}%
{%
\protected@write\@auxout{}{\string\providecommand
\string\@gls@extramakeindexopts[1]{}}
\protected@write\@auxout{}{\string\@gls@extramakeindexopts
{\@gls@extramakeindexopts}}%
}%
```

Write the name of the style file to the aux file (needed by makeglossaries)

```
\protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
\protected@write\@auxout{}{\string\@glsorder{\@glsorder}}
```

Iterate through each glossary type and activate it.

```
\@for\@glo@type:=\@glo@types\do{%
\ifthenelse{\equal{\@glo@type}{}}{}}{%
\@makeglossary{\@glo@type}}%
}%
```

New glossaries must be created before \makeglossaries so disable \newglossary.

```
\renewcommand*\newglossary[4][]{%
\PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%
```

Any subsequent instances of this command should have no effect. The deprecated \makeglossary is not redefined here as it either implements \makeglossaries or has been restored to its original definition (in which case it shouldn't be changed).

```
\let\@makeglossary\@gobble
\let\makeglossaries\relax
```

Disable all commands that have no effect after \makeglossaries

```
\@disable@onlypremakeg
```

Allow see key:

```
\let\gls@checkseeallowed\relax
```

```

Suppress warning about no \makeglossaries
  \let\warn@nomakeglossaries\relax
Activate warning about missing \printglossary
  \let\warn@noprintglossary\makeglossaries@warn@noprintglossary
Declare list parser for \glsdisplaynumberlist
  \ifglssavenumberlist
    \edef\@gls@dodolistparser{\noexpand\DeclareListParser
      {\noexpand\glsnumlistparser}{\delimN}}%
    \@gls@dodolistparser
  \fi
Prevent user from also using \makenoidxglossaries
  \let\makenoidxglossaries\@no@makeglossaries
Prohibit sort key in printgloss family:
  \renewcommand*{\@printgloss@setsort}{%
    \let\@glo@assign@sortkey\@glo@no@assign@sortkey
  }%
Check the automake setting:
  \ifglsautomake
    \renewcommand*{\@gls@doautomake}{%
      \protected@edef\gls@tmp{\@gls@automake@types}%
      \@for\@gls@type:=\gls@tmp\do{%
        \ifdefempty{\@gls@type}{}%
          {\@gls@automake{\@gls@type}}%
        }%
      }%
    \fi
Check the sort setting:
  \@glo@check@sortallowed\makeglossaries
  }%
  }
Must occur in the preamble:
  \@onlypreamble{\makeglossaries}

saries@warn@noprintglossary
\newcommand\@makeglossaries@warn@noprintglossary{%
  \ifdefstring{\@glo@types}{,}%
  {%
    \GlossariesWarningNoLine{No glossaries have been defined}%
  }%
  {%
    \GlossariesWarningNoLine{No \string\printglossary\space
      or \string\printglossaries\space
      found. ^^J(Remove \string\makeglossaries\space if you
      don't want any glossaries.) ^^JThis document will not
      have a glossary}%
  }%
}

```

```
}%
}%
```

`\glswrite` The definition of `\glswrite` has now been moved to `\makeglossaries` so that it's only defined if needed.

If `\makeglossaries` hasn't been used, issue a warning. Also issue a warning if neither `\printglossaries` nor `\printglossary` have been used.

```
\AtEndDocument{%
  \warn@nomakeglossaries
  \warn@noprintglossary
}
```

`\makenoidxglossaries` Analogous to `\makeglossaries` this activates the commands needed for `\printnoidxglossary`

```
\newcommand*\makenoidxglossaries{%
  \def\glsindexingsetting{noidx}%
  \@domakeglossaries
  {%
```

Redefine empty glossary warning:

```
\renewcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{Empty glossary for
  \string\printnoidxglossary[type=##1]}.
  Rerun may be required (or you may have forgotten to use
  commands like \string\gls)}%
}%
```

Don't escape makeindex/xindy characters:

```
\let\@gls@checkmkidxchars@gobble
```

Don't escape locations:

```
\glsesclocationsfalse
```

Write glossary information to aux instead of glossary files

```
\let\@do@wrglossary\gls@noidxglossary
```

Switch on group headings that use the character code:

```
\let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
```

Allow see key:

```
\let\gls@checkseeallowed\relax
```

Redefine cross-referencing macro:

```
\renewcommand{\@do@seeglossary}[2]{%
  \protected@edef\@gls@label{\glsdetoklabel{##1}}%
  \protected@write\@auxout{}{%
    \string\@gls@reference
    {\csname glo@\@gls@label @type\endcsname}%
    {\@gls@label}%
    {%
      \string\glsseeformat##2}%
    }%
  }%
```

```
}%
}%
```


If user removes the glossaries package from their document, ensure the next run doesn't throw a load of undefined control sequence errors when the aux file is parsed.

```

\AtBeginDocument
{%
\write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
}%

Change warning about no glossaries
\def\warn@noprintglossary{%
\GlossariesWarningNoLine{No \string\printnoidxglossary\space
or \string\printnoidxglossaries ^^J
found. (Remove \string\makenoidxglossaries\space if you
don't want any glossaries.)^^JThis document will not have a glossary}%
}%

Suppress warning about no \makeglossaries
\let\warn@nomakeglossaries\relax

Prevent user from also using \makeglossaries
\let\makeglossaries\@no@makeglossaries

Allow sort key in printgloss family:
\renewcommand*{\@printgloss@setsort}{%
\let\@glo@assign@sortkey\@glo@assign@sortkey

Initialise default sort order:
\def\@glo@sorttype{\@glo@default@sorttype}%
}%

All entries must be defined in the preamble:
\renewcommand*\new@glossaryentry[2]{%
\PackageError{glossaries}{Glossary entries must be
defined in the preamble^^Jwhen you use
\string\makenoidxglossaries}%
{Either move your definitions to the preamble or use
\string\makeglossaries}%
}%

Redefine \glsentrynumberlist
\renewcommand*\@glsentrynumberlist[1]{%
\letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
\ifdef\@gls@loclist
{%
\glsnoidxloclist{\@gls@loclist}%
}%
{%
??\glsdoifexists{##1}%
{%
\GlossariesWarning{Missing location list for '##1'. Either
a rerun is required or you haven't referenced the entry}%
}%
}%

```

```

    }%
  }%
Redefine \glsdisplaynumberlist
\renewcommand*{\glsdisplaynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo\glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \def\@gls@noidxloclist@sep{%
      \def\@gls@noidxloclist@sep{%
        \def\@gls@noidxloclist@sep{%
          \glsnumlistsep
        }%
      \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
    }%
  }%
  \def\@gls@noidxloclist@finalsep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
  \@gls@noidxloclist@finalsep
  \@gls@noidxloclist@prev
}%
{%
  ??\glsdoifexists{##1}%
  {%
    \GlossariesWarning{Missing location list for ‘##1’. Either
      a rerun is required or you haven’t referenced the entry}%
  }%
}%
}%

```

Provide a generic way of iterating through the number list:

```

\renewcommand*{\glsnumberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo\glsdetoklabel{##1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
  \let\@gls@org@glsseeformat\glsseeformat
  \let\glsnoidxdisplayloc##2\relax
  \let\glsseeformat##3\relax
  \ifdef\@gls@loclist
  {%
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
  \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
  \let\glsseeformat\@gls@org@glsseeformat

```

```

}%
Modify sanitize sort function
\let\@gls@sanitizesort\gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\gls@noidx@setsanitizesort
Check sort option allowed.
\gls@check@sortallowed\makenoidxglossaries
Implement nonumberlist
\let\@gls@savenonumberlist\gls@noidx@nonumberlist
}%
}
Preamble-only command:
\@onlypreamble{\makenoidxglossaries}

```

```
\glsnumberlistloop{<label>}{<handler>}{<xr handler cs>}
```

\glsnumberlistloop

```

\newcommand*\glsnumberlistloop}[3]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}

```

\glsnoidxnumberlistloophandler Handler macro for \glsnumberlistloop. (The argument should be in the form \glsnoidxdisplayloc{<prefix>}{<counter>}{<format>}{<n>})

```

\newcommand*\glsnoidxnumberlistloophandler}[1]{%
  #1%
}

```

\@no@makeglossaries Can't use both \makeglossaries and \makenoidxglossaries

```

\newcommand*\@no@makeglossaries{%
  \PackageError{glossaries}{You can't use both
    \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
    together.}%
}

```

\@gls@noref@warn Warning when no instances of \@gls@reference found.

```

\newcommand*\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
    is required to make \string\printnoidxglossary[type={#1}] work}%
}

```

2.14 Writing information to associated files

`\gls@noidxglossary` Write the glossary information to the aux file (for the ‘noidx’ method).

```
\newcommand*{\gls@noidxglossary}{%
  \protected@write\auxout{\glswrglossdisablelocationcmds}{%
    \string\@gls@reference
    {\csname glo@\@gls@label @type\endcsname}%
    {\@gls@label}%
    {\string\glsnoidxdisplayloc
     {\@glo@counterprefix}%
     {\@gls@counter}%
     {\@glsnumberformat}%
     {\@glslocref}%
    }%
  }%
}
```

`\istfile` Deprecated in v3.0 (2011-04-02). Removed in v4.50.

At the end of the document, the files should be created if `savewrites=true`.

```
\AtEndDocument{%
  \glswritefiles
}
```

`\@glswritefiles` Only write the files if `savewrites=true`.

```
\newcommand*{\@glswritefiles}{%
```

Iterate through all the glossaries.

```
\foralllglossaries{\@glo@type}{%
```

Check for empty glossaries (patch provided by Patrick Häcker)

```
\ifcsundef{glo@\@glo@type @filetok}%
  {%
    \def\gls@tmp{}%
  }%
  {%
    \edef\gls@tmp{\expandafter\the
      \csname glo@\@glo@type @filetok\endcsname}%
  }%
  \ifx\gls@tmp\@empty
    \ifx\@glo@type\glsdefaulttype
      \GlossariesWarningNoLine{Glossary ‘\@glo@type’ has no
        entries.^^JRemember to use package option ‘nomain’ if
you
        don’t want to^^Juse the main glossary}%
    \else
      \GlossariesWarningNoLine{Glossary ‘\@glo@type’ has no
        entries}%
    \fi
  \else
```

```

\@glsopenfile{\glswrite}{\@glo@type}%
\immediate\write\glswrite{%
  \expandafter\the
  \csname glo@\@glo@type @filetok\endcsname}%
\immediate\closeout\glswrite
\fi
}%
}

```

As from v4.10, the `\glossary` command isn't used by the `glossaries` package. Since the user isn't expected to use this command (as `glossaries` takes care of the particular format required for `makeindex/xindy`) there's no need for a user level command. Using a custom internal command prevents any conflict with other packages (and with the `\mark` mechanism).

The associated number should be stored in `\theglsentrycounter` before using `\gls@glossary`.

```

\gls@glossary
\newcommand*\gls@glossary[1]{%
  \gls@glossary{#1}%
}

```

```
\@gls@glossary{<type>}{<indexing info>}
```

```
\@gls@glossary
```

(In v4.10, `\glossary` was redefined to `\@gls@glossary` to avoid conflict with other packages.) Initially define internal `\@gls@glossary` to ignore its argument. Indexing will be enabled when `\@gls@glossary` is redefined by `\@makeglossary`.

This command was originally defined to do `\@index{<indexing info>}` so that it behaved much like `\index`. The definition was then changed to use `\index` as `memoir` changes the definition of `\@index`. (Thanks to Dan Luecking for pointing this out.)

However, if normal indexing is enabled (for example with `\makeindex`) but no glossary lists are required (so `\@makeglossary` isn't used), then `\index` will cause a problem here. The `\@index` trick allows for special characters within `<indexing info>` (so you can do, for example, `\index{%\%}`), and the original design of `\@glossary` here was actually a legacy from the old `glossary` package. With the `glossaries` package, the indexing information supplied in the second argument is more constrained and just consists of the sort value (given by the `sort key`), the actual value (given by `\glossentry{<label>}` or `\subglossentry{<level>}{<label>}`), and the format. This means that there's no need to worry about special characters appearing in the second argument as they can't be in the label or sort value. (If they are in the sort value then the category code would've needed to be changed when the entry was defined or `\glspercentchar` would be needed with the sort sanitization switched off.) This means that it's safe to simply ignore the second argument.

```

\newcommand*{\@gls@glossary}[2]{%
  \if@gls@debug
    \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
  \fi
}

```

This is a convenience command to set `\@gls@glossary`. It's used by `\@makeglossary` and then redefined to do nothing, as it only needs to be done once.

`\@gls@renewglossary`

```

\newcommand{\@gls@renewglossary}{%
  \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
  \let\@gls@renewglossary\@empty
}

```

The `\gls@wrglossary` command is defined to have two arguments. The first argument is the glossary type, the second argument is the glossary entry (the format of which is set in `\glslink`).

`\gls@wrglossary`

```

\newcommand*{\gls@wrglossary}[2]{%
  \ifgls@savewrites
    \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
    \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
      \expandafter{\@gls@tmp~J}%
  \else
    \ifcsdef{glo@#1@file}%
      {%
        \expandafter\protected@write\csname glo@#1@file\endcsname{%
          \gls@disablepagerefexpansion}{#2}%
        }%
      {%
        \ifignoredglossary{#1}{}%
          {%
            \GlossariesWarning{No file defined for glossary '#1'}%
          }%
        }%
      \fi
    \endgroup\@esphack
  }

```

`\@do@wrglossary`

```

\newcommand*{\@do@wrglossary}[1]{%
  \gls@ifnotmeasuring{\glswriteentry{#1}{\@do@wrglossary{#1}}}%
}

```

`\glswriteentry` Provide a user level command so the user can customize whether or not a line should be added to the glossary. The arguments are the label and the code that writes to the glossary file.

```

\newcommand*\glswriteentry}[2]{%
  \ifglsindexonlyfirst
    \ifglsused{#1}{#2}%
  \else
    #2%
  \fi
}

```

`\gls@protected@pagefmts` List of page formats to be protected against expansion.

```

\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}

```

`\gls@disablepagerefexpansion` As from v4.50, this will only do the page protection formats for eslocations=true.

```

\newcommand*\gls@disablepagerefexpansion){%
  \glswrglossdisablelocationcmds
  \ifglsesclocations
    \@for\@gls@this:=\gls@protected@pagefmts\do
      {%
        \expandafter\let\@gls@this\relax
      }%
  \fi
}

```

`\glswrglossdisablelocationcmds`

```

\newcommand{\glswrglossdisablelocationcmds}{}

```

`\gls@alphpage`

```

\newcommand*\gls@alphpage}{\@alph\c@page}

```

`\gls@Alphpage`

```

\newcommand*\gls@Alphpage}{\@Alph\c@page}

```

`\gls@numberpage`

```

\newcommand*\gls@numberpage}{\number\c@page}

```

`\gls@arabicpage`

```

\newcommand*\gls@arabicpage}{\@arabic\c@page}

```

`\gls@romanpage`

```

\newcommand*\gls@romanpage}{\romannumeral\c@page}

```

`\gls@Romanpage`

```

\newcommand*\gls@Romanpage}{\@Roman\c@page}

```

```

\glsaddprotectedpagefmt{<cs name>}

```

`\glsaddprotectedpagefmt`

Added a page format to the list of protected page formats. The argument should be the name (without a backslash) of the command that takes a \TeX register as the argument ($\langle csname \rangle \langle c@page \rangle$ must be valid).

```
\newcommand*\glsaddprotectedpagefmt}[1]{%
  \ifglsclocations
    \protected@eappto\gls@protected@pagefmts{\expandonce{\csname gls#1page\endcsname}}%
    \csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
    \protected@eappto\wrglossarynumberhook{%
      \noexpand\let\expandonce{\csname org@gls#1\endcsname}%
      \expandonce{\csname#1\endcsname}%
      \noexpand\def\expandonce{\csname#1\endcsname}{%
        \noexpand\@wrglossary@pageformat
          \expandonce{\csname gls#1page\endcsname}%
          \expandonce{\csname org@gls#1\endcsname}%
        }%
      }%
    \else
      \GlsWarnAddProtectedPageFmt
    \fi
}
```

$\GlsWarnAddProtectedPageFmt$

```
\newcommand{\GlsWarnAddProtectedPageFmt}{%
  \GlossariesWarning{\string\glsaddprotectedpagefmt\space has no
    effect with esclocations=false}%
}
```

$\@wrglossarynumberhook$ Hook used by $\@do@wrglossary$

```
\newcommand*\@wrglossarynumberhook{}
```

$\@wrglossary@pageformat$

```
\newcommand{\@wrglossary@pageformat}[3]{%
  \ifx#3\c@page #1\else #2#3\fi
}
```

$\@do@wrglossary$ Write the glossary entry in the appropriate format.

```
\newcommand*\@do@wrglossary}[1]{%
  \ifglsclocations
    \@do@esc@wrglossary{#1}%
  \else
    \@do@noesc@wrglossary{#1}%
  \fi
}
```

$glswrglossdisableanchorcmds$

```
\newcommand{\glswrglossdisableanchorcmds}{\let\glstexorpdfstring\@secondoftwo}
\ifdef\texorpdfstring
  {\appto\glswrglossdisableanchorcmds{\let\texorpdfstring\@secondoftwo}}{}
\ifdef\pdfstringdefPreHook
  {\appto\glswrglossdisableanchorcmds{\pdfstringdefPreHook}}{}
}
```


`\glswrglosslocationtarget` Must expand. May be used to alter the location suffix in the `\glshypernumber` anchor.

```
\newcommand{\glswrglosslocationtarget}[1]{#1}
```

`\glswrglosslocationtextfmt` Used to encapsulate the hyperlink text in `\glshypernumber`.

```
\newcommand{\glswrglosslocationtextfmt}[1]{#1}
```

`\@do@noesc@wrglossary` Write the glossary entry in the appropriate format. The locations don't need to be pre-processed before writing the information to the glossary file, but the prefix still needs to be found.

```
\newcommand*{\@do@noesc@wrglossary}[1]{%
```

Don't fully expand yet.

```
\expandafter\def\expandafter\@glslocref\expandafter{\theglentrycounter}%
\expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
```

Find the prefix if `\@glsHlocref` and `\@glslocref` aren't the same.

```
\ifx\@glsHlocref\@glslocref
\def\@glo@counterprefix{%
\else
```

The value of the counter isn't important here as it's the prefix that's of interest. (`\c@page` will have the same value in both `\theglentrycounter` and `\theHglentrycounter` at this point, even if it hasn't been updated yet. The page number is not expected to occur in the prefix.)

```
\xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
\fi
```

De-tok label if required.

```
\protected@edef\@gls@label{\glsdetoklabel{#1}}%
```

Write the information to file:

```
\@do@@wrglossary
}
```

`\ifglswrallowprimitivemods` Conditional to determine whether or not `\@do@esc@wrglossary` should be allowed to temporarily redefine `\the` and `\number`. Version 4.50 switched the default to false.

```
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodsfalse
```

`\@do@esc@wrglossary` Write the glossary entry in the appropriate format. (Need to set `\@glsnumberformat` and `\@gls@counter` prior to use.) The argument is the entry's label. This is far more complicated with `xindy` than with other indexing methods. There are two necessary but conflicting requirements with `xindy`:

1. all backslashes in the location must be escaped;
2. `\c@page` can't be prematurely expanded.

(With `makeindex` there's the remote possibility that the page compositor is a `makeindex` special character, so that would also need to be escaped.)

For example, suppose `\thepage` is defined as

```
%\renewcommand{\thepage}{\tally{page}}
%\newcommand{\tally}[1]{\tallynum{\expandafter\the\csname c@#1\endcsname}}
%
```

where `\tallynum` is a robust command that takes a number as its argument. With all indexing methods other than `xindy`, a deferred write with `\thepage` as the location will expand to `\tallynum{<n>}` where `<n>` is the page number. Since the write is deferred, the page number is correct. (`makeindex` won't accept this location format, but `\makenoidxglossaries` and `bib2gls` are quite happy with it.) Unfortunately, this fails with `xindy` because `xindy` interprets this location as `\tallynum{<n>}` because `\t` represents a the character "t". The location must be written as `\\tallynum{<n>}`.

This means that the location `\tally{page}` must be expanded and then the backslashes must be doubled. Unfortunately `\c@page` mustn't be expanded until the deferred write is performed, so the location actually needs to be expanded to `\tallynum{\the\c@page}` but the backslashes in `\the\c@page` mustn't be escaped. All other backslashes must be escaped. (In this case, only the backslash in `\tallynum` but the location format may include other control sequences.) The code below works on the assumption that commands like `\tally` are defined in the form

```
%\newcommand{\tally}[1]{\tallynum{\expandafter\the\csname c@#1\endcsname}}
%
```

(note the use of `\expandafter` and `\name`) or in the form

```
%\newcommand{\tally}[1]{\tallynum{\arabic{#1}}}}
%
```

In the second case, `\arabic` is one of the known commands that's temporarily adjusted to prevent `\c@page` from being prematurely expanded. In the first case, `\the` is temporarily modified (unless `\glswrallowprimitivemodsfalse`) to check if it's followed by `\c@page`. The `\expandafter` ensures that it is. If `\tally` is defined in another way that hides `\c@page` for example using `\the\value{#1}` then the process fails.

With `makeindex`, `\tallynum` needs to expand to just the decimal number while writing the location to the glossary file, otherwise `makeindex` will reject it. This can be done by defining `\glstallypage` so that `\tally` can locally be set to `\arabic` while expansion is occurring. Again, `\c@page` must be protected from expansion until the deferred write occurs.

The expansion before the write occurs also allows the hyper prefix to be determined where `\theH{counter}` is defined in the form `<prefix>.\the{counter}`. It's possible (although again unlikely) that a `makeindex` character might occur in the prefix, which therefore needs escaping. The prefix is passed as the

optional argument of `\setentrycounter` which is needed by commands like `\glshypernumber` to create a hyperlink for a given counter (like `\hyperpage` but for an arbitrary counter).

```
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
\begingroup
```

First a bit of hackery to prevent premature expansion of `\c@page`. Store original definitions (scoped):

```
\let\gls@orgthe\the
\let\gls@orgnumber\number
\let\gls@orgarabic\@arabic
\let\gls@orgromannumeral\romannumeral
\let\gls@orgalph\@alph
\let\gls@orgAlph\@Alph
\let\gls@orgRoman\@Roman
```

Redefine:

```
\ifglswrallowprimitivemods
```

The redefinition of `\the` to use `\expandafter` solves the problem of `\the\csname c@{counter}\endcsname` but is only a partial solution to the problem of `\the\value`. With `\value`, `\c@page` is too deeply hidden and will be expanded too soon, but at least there won't be an error.

```
\def\gls@the##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
\def\the{\expandafter\gls@the}%
\def\gls@number##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
\def\number{\expandafter\gls@number}%

\def\romannumeral##1{%
\ifx##1\c@page \gls@romanpage \else \gls@orgromannumeral##1\fi}%
\fi
\def\@arabic##1{%
\ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\@Roman##1{%
\ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
\ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
\ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
```

Add hook to allow for other number formats:

```
\@wrglossarynumberhook
```

Prevent expansion:

```
\gls@disablepagerefexpansion
```

Now store location in `\@glslocref`:

```
\protected@xdef\@glslocref{\theglentrycounter}%
\endgroup
```

Escape any special characters. It's possible that with `makeindex` the separator might be a `makeindex` special character. Although not likely, it still needs to be taken into account.

```
\@gls@checkmkidxchars\@glslocref
```

Check if the hyper-location is the same as the location and set the hyper prefix.

```
\expandafter\ifx\theHglentrycounter\theHglentrycounter\relax
\def\@glo@counterprefix{}%
\else
\protected@edef\@glsHlocref{\theHglentrycounter}%
\@gls@checkmkidxchars\@glsHlocref
\@xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
\fi
```

De-tok label if required

```
\protected@edef\@gls@label{\glsdetoklabel{#1}}%
```

Write the information to file:

```
\@do@@wrglossary
}
```

```
\@do@@wrglossary
```

```
\newcommand*{\@do@@wrglossary}{%
```

Determine whether to use `xindy` or `makeindex` syntax

```
\ifglxindy
```

Need to determine if the formatting information starts with a (or) indicating a range.

```
\expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
\def\@glo@range{}%
\expandafter\if\@glo@prefix\relax
\def\@glo@range{:open-range}%
\else
\expandafter\if\@glo@prefix\relax
\def\@glo@range{:close-range}%
\fi
\fi
```

Write to the glossary file using `xindy` syntax.

```
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
(indexentry :tkey (\csname glo@\@gls@label @index\endcsname)

:locref \string{\@glo@counterprefix}{\@glslocref}\string" %
:attr \string"\@gls@counter\@glo@suffix\string"
\@glo@range
)
}%
\else
```

Convert the format information into the format required for `makeindex`

```
\@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@gls@numberformat}%
{\@glo@counterprefix}%
```

Write to the glossary file using `makeindex` syntax.

```
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
\string\glossaryentry{\csname glo@\@gls@label @index\endcsname
\@gls@encapchar\@glo@numfmt}{\@gls@locref}}%
\fi
}
```

`\@xp@gls@getcounterprefix`

```
\newcommand*\@xp@gls@getcounterprefix[2]{%
\bgrou
\gls@rgloss@disable@anchor@cmds
\protected@edef\@do@gls@getcounterprefix{%
\noexpand\egroup
\noexpand\@gls@getcounterprefix{#1}{#2}%
}%
\@do@gls@getcounterprefix
}
```

`\@gls@getcounterprefix` Get the prefix that needs to be prepended to counter in order to get the hyper counter. (For example, with the standard article class and `hyperref`, `\theequation` needs to be prefixed with `\section num` to get the equivalent `\theHequation`.) NB this assumes that the prefix ends with a dot, which is the standard. (Otherwise it makes the xindy location classes more complicated.)

The expansion has been moved to `\@xp@gls@getcounterprefix`.

```
\newcommand*\@gls@getcounterprefix[2]{%
\def\@gls@thisloc{#1}\def\@gls@thisHloc{#2}%
\ifx\@gls@thisloc\@gls@thisHloc
\def\@glo@counterprefix{}%
\else
\def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
\def\@glo@tmp{##2}%
\ifx\@glo@tmp\@empty
\def\@glo@counterprefix{}%
\else
\def\@glo@counterprefix{##1}%
\fi
}%
\@gls@get@counterprefix#2.#1\end@getprefix
```

Warn if no prefix can be formed.

```
\ifx\@glo@counterprefix\@empty
\GlossariesWarning{Hyper target ‘#2’ can’t be formed by
prefixing location ‘#1’. You need to modify the
definition of \string\theH\@gls@counter. Jotherwise you
will get the warning: “name{\@gls@counter.#1}’ has been J
referenced but does not exist”}%
```

```

    \fi
  \fi
}

```

2.15 Glossary Entry Cross-References

`\do@seeglossary` Write the glossary entry with a cross reference. The first argument is the entry's label, the second must be in the form `[\langle tag \rangle]{\langle list \rangle}`, where `\langle tag \rangle` is a tag such as "see" and `\langle list \rangle` is a list of labels.

```

\newcommand{\do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref
\ifglsxindy
\gls@glossary{\csname glo@#1@type\endcsname}{%
(indexentry
:tkey (\csname glo@#1@index\endcsname)
:xref (\string"\@gls@xref\string")
:attr \string"see\string"
)
}%
\else
\gls@glossary{\csname glo@#1@type\endcsname}{%
\string\glossaryentry{\csname glo@#1@index\endcsname
\@gls@encapchar glsseeformat\@gls@xref}{Z}}%
\fi
}

```

`\@gls@fixbraces` If no optional argument is specified, list needs to be enclosed in a set of braces.

```

\def\@gls@fixbraces#1#2#3\@nil{%
\ifx#2[\relax
\@gls@fixbraces#1#2#3\@end@fixbraces
\else
\def#1{\{#2#3\}}%
\fi
}

```

`\@@gls@fixbraces`

```

\def\@@gls@fixbraces#1[#2]#3\@end@fixbraces{%
\def#1{\[#2\]{#3}}%
}

```

```

\glssee \glssee{\label}{\langle cross-ref list \rangle}
\newrobustcmd*{\glssee}[3][\seename]{%
\do@seeglossary{#2}{[#1]{#3}}
\newcommand*{\@glssee}[3][\seename]{%
\glssee[#1]{#3}{#2}}

```

`\glsseeformat` The first argument specifies what tag to use (e.g. “see”), the second argument is a comma-separated list of labels. The final argument (the location) is ignored.

```

\newrobustcmd*{\glsseeformat}[3][\seename]{%
  \emph{#1} \glsseelist{#2}}

```

`\glsseelist` `\glsseelist{<list>}` formats list of entry labels.

```

\newrobustcmd*{\glsseelist}[1]{%

```

If there is only one item in the list, set the last separator to do nothing.

```

  \let\@gls@dolast\relax

```

Don’t display separator on the first iteration of the loop

```

  \let\@gls@donext\relax

```

Iterate through the labels

```

  \@for\@gls@thislabel:=#1\do{%

```

Check if on last iteration of loop

```

    \ifx\@xfor@nextelement\@nnil
      \@gls@dolast
    \else
      \@gls@donext
    \fi

```

Display the entry for this label. (Expanding label as it’s a temporary control sequence that’s used elsewhere.)

```

    \expandafter\glsseeitem\expandafter{\@gls@thislabel}%

```

Update separators

```

    \let\@gls@dolast\glsseelastsep
    \let\@gls@donext\glsseesep
  }%
}

```

`\glsseelastsep` Separator to use between penultimate and ultimate entries in a cross-referencing list.

```

\newcommand*{\glsseelastsep}{\space\andname\space}

```

`\glsseesep` Separator to use between entries in a cross-referencing list.

```

\newcommand*{\glsseesep}{, }

```

`\glsseeitem` `\glsseeitem{<label>}` formats individual entry in a cross-referencing list.

```

\newrobustcmd*{\glsseeitem}[1]{\gls hyperlink[\glsseeitemformat{#1}]{#1}}

```

`\glsseeitemformat` Introduced in v3.0 to replace use of `\glsentryname` (originally in `\glsseeitem`) with `\glsentrytext` to avoid problems with the name key being sanitized. This ceased being a problem in v3.08a, but `\glsentrytext` was retained for backward-compatibility. As from v4.50, the default has reverted back to using `\glsentryname` unless the entry is an acronym (which could have a potentially long name).

```

\newcommand*{\glsseeitemformat}[1]{%
  \ifglshasshort{#1}{\glsentrytext{#1}}{\glsentryname{#1}}%
}

```

2.16 Displaying the glossary

An individual glossary is displayed in the text using `\printglossary[⟨key-val list⟩]`. If the `type` key is omitted, the default glossary is displayed. The optional argument can be used to specify an alternative glossary, and can also be used to set the style, title and entry in the table of contents. Available keys are defined below.

`\gls@save@numberlist` Provide command to store number list.

```

\newcommand*{\gls@save@numberlist}[1]{%
  \ifglssavenumberlist
    \toks@{#1}%

    \protected@edef\@do@writeaux@info{%
      \noexpand\csgdef{glo@\gls@currententrylabel @numberlist}{\the\toks@}%
    }%
    \@onelevel@sanitize\@do@writeaux@info
    \protected@write\@auxout{}{\@do@writeaux@info}%
  \fi
}
```

`\warn@noprintglossary` Warn the user if they have forgotten `\printglossaries` or `\printglossary`. (Will be suppressed if there is at least one occurrence of `\printglossary`. There is no check to ensure that there is a `\printglossary` for each defined glossary.)

```

\newcommand*{\warn@noprintglossary}{}%
```

`\printglossary` The TOC title needs to be processed in a different manner to the main title in case the translator and hyperref packages are both being used.

```

\ifcsundef{printglossary}{}%
{%
```

If `\printglossary` is already defined, issue a warning and undefine it.

```

  \gls@warnonglossdefined
  \undef\printglossary
}
```

Neither `\printglossary` nor `\printnoidxglossary` can work with an ignored glossary (since ignored glossaries normally suppress indexing and there's no associated file for `makeindex/xindy` to process). However `\printunsrtglossary` can be used with an ignored glossary, so provide a command to warn if the glossary doesn't exist or is an ignored glossary.

`\gloss@checkexists@noignored`

```

\newcommand*{\@printgloss@checkexists@noignored}[2]{%
  \ifglossaryexists{#1}%
  {#2}%
  {%
    \ifignoredglossary{#1}%
    {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
  }%
}
```



```

        {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
    }%
}

```

`ss@checkexists@allowignored` For use with `\printunsrtglossary`.

```

\newcommand*{\@printgloss@checkexists@allowignored}[2]{%
  \s@ifglossaryexists{#1}%
  {#2}%
  {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}

```

`\@printgloss@checkexists`

```

\let\@printgloss@checkexists\@printgloss@checkexists@noignored

```

`\printglossary` has an optional argument. The default value is to set the glossary type to the main glossary.

```

\newcommand*{\printglossary}[1][type=glndefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@glossary}%
}

```

The `\printglossaries` command will do `\printglossary` for each glossary type that has been defined. It is better to use `\printglossaries` rather than individual `\printglossary` commands to ensure that you don’t forget any new glossaries you may have created. It also makes it easier to chop and change the value of the acronym package option. However, if you want to list the glossaries in a different order, or if you want to set the title or table of contents entry, or if you want to use different glossary styles for each glossary, you will need to use `\printglossary` explicitly for each glossary type.

`\printglossaries`

```

\newcommand*{\printglossaries}{%
  \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}

```

`\printnoidxglossary` Provide an alternative to `\printglossary` that doesn’t require an external indexing application.

```

\newcommand*{\printnoidxglossary}[1][type=glndefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@noidx@glossary}%
}

```

`\printnoidxglossaries` Analogous to `\printglossaries`

```

\newcommand*{\printnoidxglossaries}{%
  \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}

```

`\@printgloss@setsort` Initialise to do nothing.

```

\newcommand*{\@printgloss@setsort}{}

```

`\@gls@preglossaryhook`

```
\newcommand*{\@gls@preglossaryhook}{}
```

`\@printglossary` Sets up the glossary for either `\printglossary` or `\printnoidxglossary`. The first argument is the options list, the second argument is the handler macro that deals with the actual glossary. This is also used by `glossaries-extra`'s `\printunsrtglossary` which may be used with an ignored glossary.

```
\newcommand{\@printglossary}[2]{%
```

Set up defaults.

```
\def\@glo@type{\glsdefaulttype}%
```

```
\def\glossarytitle{\csname @glo@type \@glo@type @title\endcsname}%
```

```
\def\glossarytoctitle{\glossarytitle}%
```

```
\let\org@glossarytitle\glossarytitle
```

```
\def\@glossarystyle{%
```

```
\ifx@glossary@default@style\relax
```

```
\GlossariesWarning{No default glossary style provided \MessageBreak  
for the glossary '@glo@type'. \MessageBreak
```

```
Using fallback. \MessageBreak
```

```
To fix this set the style with \MessageBreak
```

```
\string\setglossarystyle\space or use the \MessageBreak
```

```
style key=value option}%
```

```
\fi
```

```
}%
```

```
\def\gls@dotoc@title{\glssettoctitle{\@glo@type}}%
```

Store current value of `\glossaryentrynumbers`. (This may be changed via the optional argument)

```
\let\@org@glossaryentrynumbers\glossaryentrynumbers
```

Localise the effects of the optional argument

```
\gls@printglossary@begin
```

Activate or deactivate sort key:

```
\@printgloss@setsort
```

Determine settings specified in the optional argument.

```
\setkeys{printgloss}{#1}%
```

Does the glossary exist?

```
\@printgloss@checkexists{\@glo@type}%
```

```
{%
```

If title has been set, but toctitle hasn't, make toctitle the same as given title (rather than the title used when the glossary was defined)

```
\ifx\glossarytitle\org@glossarytitle
```

```
\else
```

```
\expandafter\let\csname @glo@type \@glo@type @title\endcsname
```

```
\glossarytitle
```

```
\fi
```

Allow a high-level user command to indicate the current glossary

```
\let\currentglossary\@glo@type
```

Enable individual number lists to be suppressed.

```
\let\org@glossaryentrynumbers@glossaryentrynumbers
\let\glsnonextpages@glsnonextpages
```

Enable individual number list to be activated:

```
\let\glsnextpages@glsnextpages
```

Enable suppression of description terminators.

```
\let\nopostdesc@nopostdesc
```

Set up the entry for the TOC

```
\gls@dotoc@title
```

Set the glossary style

```
\@glossarystyle
```

Added a way to fetch the current entry label (v3.08 updated for new `\glossentry` and `\subglossentry`, but this is now only needed for backward compatibility):

```
\let\gls@org@glossaryentryfield@glossentry
\let\gls@org@glossarysubentryfield@subglossentry
\renewcommand{\glossentry}[1]{%
  \protected@xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
  \gls@org@glossaryentryfield{##1}%
}%
\renewcommand{\subglossentry}[2]{%
  \protected@xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
  \gls@org@glossarysubentryfield{##1}{##2}%
}%
```

```
\@gls@preglossaryhook
```

Now do the handler macro that deals with the actual glossary:

```
#2%
}%
```

End the current scope

```
\gls@printglossary@end
```

Reset `\glossaryentrynumbers`

```
\global\let@glossaryentrynumbers@org@glossaryentrynumbers
```

Suppress warning about no `\printglossary`

```
\global\let@warn@noprintglossary@relax
}
```

```
\gls@printglossary@begin
```

```
\newcommand{\gls@printglossary@begin}{\begingroup}
```

`\gls@printglossary@end` See <https://github.com/latex3/tagging-project/issues/48>

```
\newcommand{\gls@printglossary@end}{%
\expandafter\endgroup\if@endpe\doendpe\fi
}
```

`\@print@glossary` Internal workings of `\printglossary` dealing with reading the external file.

```
\newcommand{\@print@glossary}{%
```

Some macros may end up being expanded into internals in the glossary, so need to make `@` a letter. (Unlikely to be a problem since v3.08a but kept for backward compatibility.)

```
\makeatletter
```

Input the glossary file, if it exists.

```
\input@{\jobname.\csname @glo@type@\glo@type @in\endcsname}%
```

If the glossary file doesn't exist, do `\null`. (This ensures that the page is shipped out and all write commands are done.) This might produce an empty page, but at this point the document isn't complete, so it shouldn't matter.

```
\IfFileExists{\jobname.\csname @glo@type@\glo@type @in\endcsname}%
{}%
{\null}%
```

If `xindy` is being used, need to write the language dependent information to the `.aux` file for `makeglossaries`.

```
\ifglsxindy
```

Set `\@gls@langmod`:

```
\gls@xdy@set@lang{\glo@type}%
\edef\@do@auxoutstuff{%
\noexpand\AtEndDocument{%
```

If the user removes the glossary package from their document, ensure the next run doesn't throw a load of undefined control sequence errors when the aux file is parsed.

```
\noexpand\immediate\noexpand\write\@auxout{%
\string\providecommand\string\@xdy@language[2]{}%
\noexpand\immediate\noexpand\write\@auxout{%
\string\@xdy@language{\glo@type}{\@gls@langmod}}%
}%
}%
\@do@auxoutstuff
\edef\@do@auxoutstuff{%
\noexpand\AtEndDocument{%
```

If the user removes the glossaries package from their document, ensure the next run doesn't throw a load of undefined control sequence errors when the aux file is parsed.

```
\noexpand\immediate\noexpand\write\@auxout{%
\string\providecommand\string\@gls@codepage[2]{}%
\noexpand\immediate\noexpand\write\@auxout{%
```

```

        \string\@gls@codepage{\@glo@type}{\gls@codepage}}%
    }%
  }%
  \@do@auxoutstuff
\fi

```

Activate warning if `\makeglossaries` hasn't been used.

```

\renewcommand*{\@warn@nomakeglossaries}{%
  \GlossariesWarningNoLine{\string\makeglossaries\space
    hasn't been used,^^Jthe glossaries will not be updated}%
}%
}

```

The sort macros all have the syntax:

```
\@glo@sortmacro@⟨order⟩{⟨type⟩}
```

where $\langle order \rangle$ is the sort order as specified by the sort key and $\langle type \rangle$ is the glossary type. (The referenced entry list is stored in $\@glsref@⟨type⟩$). The actual sorting is done by $\@glo@sortentries{⟨handler⟩}{⟨type⟩}$.

New to v4.55: Check if `datatool-base v3.0+` is installed, which provides the command `\datatool_sortwordseq:NN`

```

\ExplSyntaxOn
\cs_if_exist:NTF \datatool_sortwordseq:NN
{

```

Sequence variable in which to store entries that need sorting.

```
\seq_new:N \l__glossaries_actual_sort_seq
```

Actual sequence uses markup to expand to either the label or the provided sort value. This is to assist `datatool-base` localisation group handlers.

```

\cs_new:Nn \__glossaries_actual_sort:nn
{
  \exp_not:n { #1 }
}
\cs_new:Nn \__glossaries_actual_label:nn
{
  \exp_not:n { #2 }
}
\cs_new:Nn \__glossaries_actual:nn
{
  \__glossaries_actual_sort:nn { #1 } { #2 }
}

```

Sequence variable in which to store labels of entries that need sorting.

```
\seq_new:N \l__glossaries_sort_seq
```

Mapping from label to sort value.

```
\prop_new:N \l__glossaries_sort_prop
```

Scratch variable.

```
\tl_new:N \l__glossaries_sort_value_tl
\tl_new:N \l__glossaries_entry_label_tl
```

Information message:

```
\msg_new:nnn { glossaries } { presort }
{
  Preparing ~ to ~ sort ~ glossary ~ '#1' ~
  (total ~ number ~ of ~ referenced ~ entries: ~ #2 )
}
```

Set the sequence from the glossary list.

```
\cs_new:Nn \__glossaries_set_sort_seq:n
{
  \msg_note:nnee { glossaries } { presort }
  { #1 }
  { \__glossaries_glsref_count:n { #1 } }
  \seq_clear:N \l__glossaries_sort_seq
  \seq_clear:N \l__glossaries_actual_sort_seq
  \prop_clear:N \l__glossaries_sort_prop
  \forlistcsloop
  \__glossaries_add_sort_item:n
  { @glsref@ #1 }
}
```

Add label to the sequence if not already added.

```
\cs_new:Nn \__glossaries_add_sort_item:n
{
  \seq_if_in:NnF \l__glossaries_sort_seq { #1 }
  {
    \ifglshasparent { #1 }
    {
```

If this entry has a parent, ensure that the parent is also added. This uses recursion to ensure that all ancestors are added.

```
\exp_args:Nv \__glossaries_add_sort_item:n
{ glo@ #1 @parent }
```

The parent's hierarchical sort value should now be stored in the property list.

```
\prop_get:NvN \l__glossaries_sort_prop
{ glo@ #1 @parent }
\l__glossaries_sort_value_tl
\quark_if_no_value:NTF
\l__glossaries_sort_value_tl
{
```

Shouldn't happen, but if it does then assume no parent.

```
\tl_clear:N \l__glossaries_sort_value_tl
}
{
```

Append marker to ensure that child entries are placed immediately after parent entry.

```

        \tl_put_right:Nn \l__glossaries_sort_value_tl
        { \datatoolctrlboundary \datatoolasciistart }
    }
}

```

No parent, so no hierarchical sort prefix.

```

\tl_clear:N \l__glossaries_sort_value_tl
}

```

Append the entry's sort value to ensure hierarchy preserved, if applicable.

```

\tl_put_right:Nv \l__glossaries_sort_value_tl
{ glo@ #1 @sort }

```

Add the hierarchical sort value to the property list.

```

\prop_put:NnV \l__glossaries_sort_prop { #1 }
\l__glossaries_sort_value_tl

```

Add this entry's label to the sequence.

```

\seq_put_right:Nn \l__glossaries_sort_seq { #1 }

```

Add both the sort value and label to the actual sequence.

```

\seq_put_right:Ne \l__glossaries_actual_sort_seq
{
  \exp_not:N \__glossaries_actual:nn
  { \exp_not:V \l__glossaries_sort_value_tl }
  { #1 }
}
}
}

```

Fetch the sort value from the property map:

```

\cs_new:Nn \__glossaries_fetch_sort:n
{
  \group_begin:
  \cs_set_eq:NN \__glossaries_actual:nn \__glossaries_actual_label:nn
  \exp_args:NNNe
  \group_end:
  \tl_set:Nn \l__glossaries_entry_label_tl { #1 }
  \prop_get:NVN \l__glossaries_sort_prop
  \l__glossaries_entry_label_tl
  \l__glossaries_sort_value_tl
}

```

Available sort methods. Word sort:

```

\cs_new:Nn \__glossaries_sort_word:n
{
  \__glossaries_do_sort:Nn
  \__glossaries_sort_word_hander:nN
  { #1 }
}

```

```

}
\cs_new:Nn \__glossaries_sort_word_hander:nN
{
  \__glossaries_fetch_sort:n { #1 }
  \exp_args:NV \DTLsortwordhandler
  \l__glossaries_sort_value_tl #2
}

```

Letter sort:

```

\cs_new:Nn \__glossaries_sort_letter:n
{
  \__glossaries_do_sort:Nn
  \__glossaries_sort_letter_hander:nN
  { #1 }
}
\cs_new:Nn \__glossaries_sort_letter_hander:nN
{
  \__glossaries_fetch_sort:n { #1 }
  \exp_args:NV \DTLsortletterhandler
  \l__glossaries_sort_value_tl #2
}

```

Case-sensitive.

```

\cs_new:Nn \__glossaries_sort_case:n
{
  \__glossaries_do_sort:Nn
  \__glossaries_sort_case_hander:nN
  { #1 }
}

```

Closest match for original case-sensitive sorting. This omits the localisation support.

```

\cs_new:Nn \__glossaries_sort_case_hander:nN
{
  \__glossaries_fetch_sort:n { #1 }
  \exp_args:NV \DTLsortwordcasehandler
  \l__glossaries_sort_value_tl #2
  \datatool_sort_handler_preprocess:Nn
  #2 { \text_lowercase:n { \l__glossaries_sort_value_tl } }
  \tl_set:Ne #2 { \text_purify:n { #2 } \datatoolctrlboundary }
}

```

Case-insensitive.

```

\cs_new:Nn \__glossaries_sort_nocase:n
{
  \__glossaries_do_sort:Nn
  \__glossaries_sort_nocase_hander:nN
  { #1 }
}

```

Closest match for original case-insensitive sorting. This omits the localisation support.


```

\cs_new:Nn \__glossaries_sort_nocase_hander:nN
{
  \__glossaries_fetch_sort:n { #1 }
  \exp_args:NV \DTLsortwordcasehandler
    \l__glossaries_sort_value_tl #2
  \datatool_sort_handler_preprocess:Nn
    #2 { \l__glossaries_sort_value_tl }
  \tl_set:Ne #2 { \text_purify:n { #2 } \datatoolctrlboundary }
}

```

Syntax: $\langle handler-cs \rangle \{ \langle type \rangle \}$

```

\cs_new:Nn \__glossaries_do_sort:Nn
{
  \renewcommand \dtlnonlettergroup [ 1 ] { glssymbols }
  \renewcommand \dtlnumbergroup [ 1 ] { glsnumbers }
  \__glossaries_set_sort_seq:n { #2 }
  \datatool_sortwordseq:NN
    \l__glossaries_actual_sort_seq
    #1
  \tl_clear:c { @glsref@ #2 }
  \seq_map_inline:Nn \l__glossaries_actual_sort_seq
    {
      \tl_set:Ne \l__glossaries_entry_label_tl { ##1 }
      \group_begin:
        \cs_set_eq:NN \__glossaries_actual:nm \__glossaries_actual_label:nm
        \exp_args:NNNe
          \group_end:
          \tl_set:Nn \l__glossaries_entry_label_tl
            { \l__glossaries_entry_label_tl }
        \exp_args:NnV \listcseadd { @glsref@ #2 } \l__glossaries_entry_label_tl
        \tl_set:ce
          { glo@ \l__glossaries_entry_label_tl @sort }
          { ##1 }
    }
}

```

Fetch the letter group from the sort value:

```

\cs_new:Npn \@gls@fetch@lettergrp
{
  \tl_if_head_eq_meaning:VNTF \@gls@sort \__datatool_sorted:nm
  {
    \tl_set:Nx \@glo@thislettergrp
      { \exp_args:NV \DTLsortedletter \@gls@sort }
  }
  {
    \exp_args:NVV \DTLassignlettergroup \@gls@sort \@gls@sort
      \@glo@thislettergrp
  }
}

```

Old version of datatool-base, so use old method.

```

\cs_new:Nn \__glossaries_sort_word:n
{
  \glo@sortentries{\glo@sorthandler@word}{#1}
}
\cs_new:Nn \__glossaries_sort_letter:n
{
  \glo@sortentries{\glo@sorthandler@letter}{#1}
}
\cs_new:Nn \__glossaries_sort_case:n
{
  \glo@sortentries{\glo@sorthandler@case}{#1}
}
\cs_new:Nn \__glossaries_sort_nocase:n
{
  \glo@sortentries{\glo@sorthandler@nocase}{#1}
}
\cs_new:Npn \gls@fetch@lettergrp
{
  \expandafter\glo@grabfirst\gls@sort{}{\}\@nil
}
}

```

Provide command that can be used once L3 syntax switched off to ease conversion. (I don't currently have time to rewrite the entire package.)

```

\cs_new:Npn \@glo@sortentries #1 #2
{
  \cs_if_exist:cTF { __glossaries_sort_ #1 :n }
  {
    \use:c { __glossaries_sort_ #1 :n } { #2 }
  }
  {
    \PackageError { glossaries }
    { Unknown ~ sort ~ method ~ '#1' } { }
  }
}
\ExplSyntaxOff

```

```

\glo@sortentries Syntax: \handler-cs{<type>}
\newcommand*{\glo@sortentries}[2]{%
  \glosortentrieswarning
  \def\glo@sortinglist{}%
  \def\glo@sortinghandler{#1}%

  \protected@edef\glo@type{#2}%
  \forlistcsloop{\glo@do@sortentries}{\glsref@#2}%
  \csdef{\glsref@#2}{}%
  \for\@this@label:=\glo@sortinglist\do{%

```

Has this entry already been added?

```

\ifinlistcs{\@this@label}{@glsref@#2}%
{}%
{%
  \listcsxadd{@glsref@#2}{\@this@label}%
}%
\ifcsdef{@glo@sortingchildren@\@this@label}%
{%
  \glo@addchildren{#2}{\@this@label}%
}%
{}%
}%
}

```

```
\@glo@addchildren{<type>}{<parent>}
```

\@glo@addchildren

```

\newcommand*{\@glo@addchildren}[2]{%
Scope to allow nesting.
  \bgroup
  \letcs{\@glo@childlist}{@glo@sortingchildren@#2}%
  \for\@this@childlabel:=\@glo@childlist\do
  {%

```

Check this label hasn't already been added.

```

  \ifinlistcs{\@this@childlabel}{@glsref@#1}%
  {}%
  {%
    \listcsxadd{@glsref@#1}{\@this@childlabel}%
  }%

```

Does this child have children?

```

  \ifcsdef{@glo@sortingchildren@\@this@childlabel}%
  {%
    \@glo@addchildren{#1}{\@this@childlabel}%
  }%
  {%
  }%
  }%
\egroup
}

```

\@glo@do@sortentries

```

\newcommand*{\@glo@do@sortentries}[1]{%
  \ifglshasparent{#1}%
  {%

```

This entry has a parent, so add it to the child list

```

  \protected@edef\@glo@parent{\csuse{glo\glsdetoklabel{#1}@parent}}%
  \ifcsundef{@glo@sortingchildren@\@glo@parent}%
  {%

```

```

        \csdef{@glo@sortingchildren@{@glo@parent}{}}%
    }%
    {}%
    \expandafter@glo@sortedinsert
        \csname @glo@sortingchildren@{@glo@parent}\endcsname{#1}%
Has the parent been added?
    \xifinlistcs{@glo@parent}{@glsref@{@glo@type}}%
    {%
Yes, it has so do nothing.
    }%
    {%
No, it hasn't so add it now.
        \expandafter@glo@do@sortentries\expandafter{@glo@parent}%
    }%
    }%
    {%
        \glo@sortedinsert{\glo@sortinglist}{#1}%
    }%
}

```

`\glo@sortedinsert{<list>}{<entry label>}`

`\glo@sortedinsert`

Insert into list.

```

\newcommand*{@glo@sortedinsert}[2]{%
    \dtl@insertinto{#2}{#1}{@glo@sortinghandler}%
}%

```

The sort handlers need to be in the form required by datatool's `\dtl@sortlist` macro. These must set the count register `\dtl@sortresult` to either `-1` (`#1` less than `#2`), `0` (`#1 = #2`) or `+1` (`#1` greater than `#2`).

`\glo@sorthandler@word`

```

\newcommand*{@glo@sorthandler@word}[2]{%
    \letcs@gls@sort@A{glo@glsdetoklabel{#1}@sort}%
    \letcs@gls@sort@B{glo@glsdetoklabel{#2}@sort}%
    \edef@glo@do@compare{%
        \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
        {\expandonce@gls@sort@B}%
        {\expandonce@gls@sort@A}%
    }%
    \glo@do@compare
}

```

`\glo@sorthandler@letter`

```

\newcommand*{@glo@sorthandler@letter}[2]{%
    \letcs@gls@sort@A{glo@glsdetoklabel{#1}@sort}%
}

```

```

\letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
\edef\glo@do@compare{%
  \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
  {\expandonce\@gls@sort@B}%
  {\expandonce\@gls@sort@A}%
}%
\glo@do@compare
}

```

\@glo@sorthandler@case Case-sensitive sort.

```

\newcommand*{\@glo@sorthandler@case}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}

```

\@glo@sorthandler@nocase Case-insensitive sort.

```

\newcommand*{\@glo@sorthandler@nocase}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlicompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}

```

\@glo@sortmacro@word Sort macro for ‘word’

```

\newcommand*{\@glo@sortmacro@word}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{word}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=word]}{ }%
  }%
}

```

\@glo@sortmacro@letter Sort macro for ‘letter’

```

\newcommand*{\@glo@sortmacro@letter}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%

```

```

    {%
      \@glo@sortentries{letter}{#1}%
    }%
    {%
      \PackageError{glossaries}{Conflicting sort options:^^J
        \string\usepackage[sort=\glo@default@sorttype]{glossaries}^^J
        \string\printnoidxglossary[sort=letter]}{ }%
    }%
  }

```

`\glo@sortmacro@standard` Sort macro for ‘standard’. (Use either ‘word’ or ‘letter’ order.)

```

\newcommand*\@glo@sortmacro@standard[1]{%
  \ifdefstring{\glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\glsorder}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=standard]}{ }%
  }%
}

```

`\glo@sortmacro@case` Sort macro for ‘case’

```

\newcommand*\@glo@sortmacro@case[1]{%
  \ifdefstring{\glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{case}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=case]}{ }%
  }%
}

```

`\glo@sortmacro@nocase` Sort macro for ‘nocase’

```

\newcommand*\@glo@sortmacro@nocase[1]{%
  \ifdefstring{\glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{nocase}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=nocase]}{ }%
  }%
}

```

`\@glo@sortmacro@def` Sort macro for ‘def’. The order of definition is given in `\glo@list@{type}`.

```
\newcommand*{\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{}%
  \forlseq@sentries[#1]{\@gls@thislabel}%
  {%
    \xifinlistcs{\@gls@thislabel}{\@glsref@#1}%
    {%
      \listadd{\@glo@sortinglist}{\@gls@thislabel}%
    }%
  }%
  Hasn't been referenced.
  }%
}%
\cslet{\@glsref@#1}{\@glo@sortinglist}%
}
```

`\@glo@sortmacro@def@do` This won't include parent entries that haven't been referenced.

```
\newcommand*{\@glo@sortmacro@def@do}[1]{%
  \ifinlistcs{#1}{\@glsref@\@glo@type}%
  {}%
  {%
    \listcsadd{\@glsref@\@glo@type}{#1}%
  }%
  \ifcsdef{\@glo@sortingchildren@#1}%
  {%
    \@glo@addchildren{\@glo@type}{#1}%
  }%
  {}%
}
```

`\@glo@sortmacro@use` Sort macro for ‘use’. (No sorting is required, as the entries are already in order of use, so do nothing.)

```
\newcommand*{\@glo@sortmacro@use}[1]{}
```

`\@print@noidx@glossary` Glossary handler for `\printnoidx@glossary` which doesn't use an indexing application. Since `\printnoidx@glossary` may occur at the start of the document, we can't just check if an entry has been used. Instead, the first pass needs to write information to the aux file every time an entry is referenced. This needs to be read in on the second run and stored in a list corresponding to the appropriate glossary.

```
\newcommand*{\@print@noidx@glossary}{%
  \ifcsdef{\@glsref@\@glo@type}%
  {%
```

Sort the entries:

```
\ifcsdef{\@glo@sortmacro@\@glo@sorttype}%
{%
  \csuse{\@glo@sortmacro@\@glo@sorttype}{\@glo@type}%
}
```

```

}%
{%
  \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}%
}%

```

Do the glossary heading and preamble

```

\glossarysection[\glossarytoctitle]{\glossarytitle}%
\glossarypreamble

```

The glossary style might use a tabular-like environment, which may cause scoping problems when setting the current letter group. The predefined tabular-like styles don't support letter group headings, but there's nothing to stop the user from defining their own custom style that might, so any redefinition of this command within theglossary will have to be done globally.

```

\def\@gls@currentlettergroup{}%
\begin{theglossary}%
\glossaryheader
\glsresetentrylist

```

Iterate through the entries.

```

\forlistcsloop{\@gls@noidx@do}{\@glsref@\@glo@type}%

```

Finally end the glossary and do the postamble:

```

\end{theglossary}%
\glossarypostamble
}%
{%
  \@gls@noref@warn{\@glo@type}%
}%
}

```

`\glo@grabfirst` This won't work for UTF-8 and inputenc.

```

\def\glo@grabfirst#1#2\@nil{%
  \def\@gls@firsttok{#1}%
  \ifdefempty\@gls@firsttok
  {%
    \def\@glo@thislettergrp{0}%
  }%
  {%

```

Sanitize it:

```

  \@onelevel@sanitize\@gls@firsttok

```

Fetch the first letter:

```

  \expandafter\@glo@grabfirst\@gls@firsttok{}{}\@nil
}%
}

```

`\@glo@grabfirst`

```

\def\@glo@grabfirst#1#2\@nil{%
  \ifdefempty\@glo@thislettergrp

```



```

{%
  \def\@glo@thislettergrp{glssymbols}%
}%
{%
  \count@=\uccode‘#1\relax
  \ifnum\count@=0\relax
    \def\@glo@thislettergrp{glssymbols}%
  \else
    \ifdefstring\@glo@sorttype{case}%
    {%
      \count@=#1\relax
    }%
    {%
    }%
  }%
  \edef\@glo@thislettergrp{\the\count@}%
\fi
}%
}

```

`\@gls@noidx@do` Handler for list iteration used by `\@print@noidx@glossary`. The argument is the entry label. This only allows one sublevel.

```
\newcommand{\@gls@noidx@do}[1]{%
```

Get this entry’s location list

```
\global\letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
```

Does this entry have a parent?

```
\ifglshasparent{#1}%
{%
```

Has a parent.

```

\gls@level=\csuse{glo@\glsdetoklabel{#1}@level}\relax
\ifdefvoid{\@gls@loclist}
{%
  \subglossentry{\gls@level}{#1}{}%
}%
{%
  \subglossentry{\gls@level}{#1}%
  {%
    \glsnoidxprenumberlist{#1}%
    \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
  }%
}%
}%

```

Doesn’t have a parent Get this entry’s sort key

```
\letcs{\@gls@sort}{glo@\glsdetoklabel{#1}@sort}%
```

Get the letter group:

```
\@gls@fetch@lettergrp
```

```

\ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
{}%
{%

```

Do the group header:

```

\ifdefempty{\@gls@currentlettergroup}{}%
{%

```

The group skip may start a new scope, so make a global assignment.

```

\global\let\@glo@thislettergrp\@glo@thislettergrp
\glsgroupskip
}%
\glsgroupheading{\@glo@thislettergrp}%
}%

```

```

\global\let\@gls@currentlettergroup\@glo@thislettergrp

```

Do this entry:

```

\ifdefvoid{\@gls@loclist}
{%
\glossentry{#1}{}%
}%
{%
\glossentry{#1}%
{%
\glsnoidxprenumberlist{#1}%
\glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
}%
}%
}%
}

```

`\glsnoidxloclist`

```
\glsnoidxloclist{<list cs>}
```

Display location list.

```

\newcommand*\glsnoidxloclist[1]{%
\def\@gls@noidxloclist@sep{}%
\def\@gls@noidxloclist@prev{}%
\forlistloop{\glsnoidxloclisthandler}{#1}%
}

```

`\glsnoidxloclisthandler` Handler for location list iterator.

```

\newcommand*\glsnoidxloclisthandler[1]{%
\ifdefstring{\@gls@noidxloclist@prev}{#1}%
{%
Same as previous location so skip.
}%
{%
\@gls@noidxloclist@sep

```

```

    #1%
    \def\@gls@noidxloclist@sep{\delimN}%
    \def\@gls@noidxloclist@prev{#1}%
  }%
}

```

`\glsnoidxdisplayloclisthandler` Handler for location list iterator when used with `\glsdisplaynumberlist`.

```

\newcommand*\glsnoidxdisplayloclisthandler[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%

```

Same as previous location so skip.

```

  }%
  {%
    \@gls@noidxloclist@sep
    \@gls@noidxloclist@prev
    \def\@gls@noidxloclist@prev{#1}%
  }%
}

```

```

\glsnoidxdisplayloc{<prefix>}{<counter>}{<format>}
  {<location>}

```

`\glsnoidxdisplayloc`

Display a location in the location list.

```

\newcommand*\glsnoidxdisplayloc[4]{%
  \setentrycounter[#1]{#2}%
  \csuse{#3}{#4}%
}

```

```

\@gls@reference{<type>}{<label>}{<loc>}

```

`\@gls@reference`

Identifies that a reference has been used (for use in the aux file). All entries must be defined in the preamble.

```

\ExplSyntaxOn
\newcommand*\@gls@reference}[3]{%

```

Add to label list

```

  \glsdoifexistsorwarn{#2}%
  {%
    \__glossaries_add_glsref:nn { #1 } { #2 }
  }%

```

Add to location list

```

  \ifcsundef{glo@glstdetoklabel{#2}@loclist}%
  {\csgdef{glo@glstdetoklabel{#2}@loclist}{}}%
  {}%
  \listcsgadd{glo@glstdetoklabel{#2}@loclist}{#3}%
}
}

```

This may be updated in future to use a sequence variable instead.

```

\cs_new:Nn \__glossaries_add_glsref:nn
{
  \ifcsundef { @glsref@ #1 }
  {
    \csgdef { @glsref@ #1 } { }
    \int_new:c { g__glossaries_gls_ref_ #1 _int }
  }
  { }
  \ifinlistcs { #2 } { @glsref@ #1 }
  { }
  {
    \listcsgadd { @glsref@ #1 } { #2 }
    \int_gincr:c { g__glossaries_gls_ref_ #1 _int }
  }
}
\cs_new:Nn \__glossaries_glsref_count:n
{
  \int_if_exist:cTF { g__glossaries_gls_ref_ #1 _int }
  {
    \int_use:c { g__glossaries_gls_ref_ #1 _int }
  }
  {
    \c_zero_int
  }
}
\ExplSyntaxOff

```

The keys that can be used in the optional argument to `\printglossary` or `\printnoidxglossary` are as follows: The `type` key sets the glossary type.

```
\define@key{printgloss}{type}{\def\@glo@type{#1}}
```

The `title` key sets the title used in the glossary section header. This overrides the title used in `\newglossary`.

```

\define@key{printgloss}{title}{%
  \def\glossarytitle{#1}%
  \let\gls@dotoc@title\relax
}

```

The `toctitle` sets the text used for the relevant entry in the table of contents.

```

\define@key{printgloss}{toctitle}{%
  \def\glossarytoctitle{#1}%
  \let\gls@dotoc@title\relax
}

```

The `style` key sets the glossary style (but only for the given glossary).

```

\define@key{printgloss}{style}{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}%
    {Glossary style ‘#1’ undefined}{%

```

```

}%
{%
  \def\@glossarystyle{\setglossarystyle{#1}}%
}%
}

```

The `numberedsection` key determines if this glossary should be in a numbered section.

```

\define@choicekey{printgloss}{numberedsection}%
  [\gls@numberedsection@val\gls@numberedsection@nr]%
  {false,nolabel,autolabel,nameref}[nolabel]%
{%
  \ifcase\gls@numberedsection@nr\relax
    \renewcommand*\@glossarysecstar{*}%
    \renewcommand*\@glossaryseclabel{}%
  \or
    \renewcommand*\@glossarysecstar{}%
    \renewcommand*\@glossaryseclabel{}%
  \or
    \renewcommand*\@glossarysecstar{}%
    \renewcommand*\@glossaryseclabel{\label{\glsautoprefix\@glo@type}}%
  \or
    \renewcommand*\@glossarysecstar{*}%
    \renewcommand*\@glossaryseclabel{%
      \protected@edef\@currentlabelname{\glossarytoctitle}%
      \label{\glsautoprefix\@glo@type}}%
  \fi
}

```

The `nogroupskip` key determines whether or not there should be a vertical gap between glossary groups.

```

\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
  \csuse{glsnogroupskip#1}%
}

```

The `nopostdot` key has the same effect as the package option of the same name.

```

\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
  \csuse{glsnopostdot#1}%
}

```

`\GlsEntryCounterLabelPrefix` Make it easier to redefine the label prefix.

```

\newcommand*\GlsEntryCounterLabelPrefix{\glsentry-}

```

The conditionals have been moved inside the appropriate commands to make it easier for the user to redefine them in the preamble and selectively switch the counter display on and off. Previously the helper commands were redefined by the `entrycounter` option, which would counteract any earlier customisation.

The `entrycounter` key is the same as the package option but localised to the current glossary.

```

\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
  \csuse{glentrycounter#1}%
  \@gls@define@glossaryentrycounter
}

```

The `subentrycounter` key is the same as the package option but localised to the current glossary. Note that this doesn't affect the parent/child counter attributes, which occurs if `subentrycounter` and `entrycounter` package options are set to true.

```

\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
  \csuse{glssubentrycounter#1}%
  \@gls@define@glossarysubentrycounter
}

```

The `nonumberlist` key determines if this glossary should have a number list.

```

\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
\ifglsnonumberlist
  \def\glossaryentrynumbers##1{%
\else
  \def\glossaryentrynumbers##1{##1}%
\fi}

```

The `sort` key sets the glossary sort handler (`\printnoidxglossary` only).

```

\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}

```

`\@glo@no@assign@sortkey` Issue error if used with `\printglossary`

```

\newcommand*{\@glo@no@assign@sortkey}[1]{%
  \PackageError{glossaries}{‘sort’ key not permitted with
\string\printglossary}%
  {The ‘sort’ key may only be used with \string\printnoidxglossary}%
}

```

`\@@glo@assign@sortkey` For use with `\printnoidxglossary`

```

\newcommand*{\@@glo@assign@sortkey}[1]{%
  \def\@glo@sorttype{#1}%
}

```

`\@glsnonextpages` Suppresses the next number list only. Global assignments required as it may not occur in the same level of grouping as the next numberlist. (For example, if `\glsnonextpages` is place in the entry's description and 3 column tabular style glossary is used.) `\org@glossaryentrynumbers` needs to be set at the start of each glossary, in the event that `\glossaryentrynumber` is redefined.

```

\newcommand*{\@glsnonextpages}{%
  \gdef\glossaryentrynumbers##1{%
    \glsresetentrylist
  }%
}

```

`\@glsnextpages` Activate the next number list only. Global assignments required as it may not occur in the same level of grouping as the next numberlist. (For example, if

`\glsnextpages` is placed in the entry's description and 3 column tabular style glossary is used.) `\org@glossaryentrynumbers` needs to be set at the start of each glossary, in the event that `\glossaryentrynumber` is redefined.

```
\newcommand*{\glsnextpages}{%
  \gdef\glossaryentrynumbers##1{%
    ##1\glsresetentrylist}}
```

`\glsresetentrylist` Resets `\glossaryentrynumbers`

```
\newcommand*{\glsresetentrylist}{%
  \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
```

`\glsnonextpages` Outside of `\printglossary` this does nothing.

```
\newcommand*{\glsnonextpages}{}
```

`\glsnextpages` Outside of `\printglossary` this does nothing.

```
\newcommand*{\glsnextpages}{}
```

Process `entrycounter` and then `subentrycounter` options (this ensures the sub-counter can pick up the main counter as the master if required):

```
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
```

`\glsresetsubentrycounter` Resets the `glossarysubentry` counter.

```
\newcommand*{\glsresetsubentrycounter}{%
  \ifglssubentrycounter
    \setcounter{glossarysubentry}{0}%
  \fi
}
```

`\glsresetentrycounter` Resets the `glossaryentry` counter.

```
\newcommand*{\glsresetentrycounter}{%
  \ifglsentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
```

`\glsstepentry` Advance the `glossaryentry` counter if in use. The argument is the label associated with the entry.

```
\newcommand*{\glsstepentry}[1]{%
  \ifglsentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
```

`\glsstepsubentry` Advance the `glossarysubentry` counter if in use. The argument is the label associated with the subentry.

```
\newcommand*{\glsstepsubentry}[1]{%
  \ifglssubentrycounter
```

```

        \protected@edef\currentglssubentry{\glsdetoklabel{#1}}%
        \refstepcounter{glossarysubentry}%
        \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
    \fi
}

```

`\glsrefentry` Reference the entry or sub-entry counter if in use, otherwise just do `\gls`.

```

\newcommand*\glsrefentry}[1]{%
  \ifglentrycounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else
    \ifglssubentrycounter
      \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
      \gls{#1}%
    \fi
  \fi
}

```

`\glssentrycounterlabel` Defines how to display the glossaryentry counter.

```

\newcommand*\glssentrycounterlabel}{%
  \ifglentrycounter
    \theglossaryentry.\space
  \fi
}

```

`\glsssubentrycounterlabel` Defines how to display the glossarysubentry counter.

```

\newcommand*\glsssubentrycounterlabel}{%
  \ifglsssubentrycounter
    \theglossarysubentry)\space
  \fi
}

```

`\glsentryitem` Step and display glossaryentry counter, if appropriate.

```

\newcommand*\glsentryitem}[1]{%
  \ifglentrycounter
    \glsstepentry{#1}\glssentrycounterlabel
  \else
    \glsresetsubentrycounter
  \fi
}

```

`\glsssubentryitem` Step and display glossarysubentry counter, if appropriate.

```

\newcommand*\glsssubentryitem}[1]{%
  \ifglsssubentrycounter
    \glsstepsubentry{#1}\glsssubentrycounterlabel
  \fi
}

```


`theglossary` (*env.*) If the `theglossary` environment has already been defined, a warning will be issued. This environment should be redefined by glossary styles.

```
\ifcsundef{theglossary}%
{%
  \newenvironment{theglossary}{}{}%
}%
{%
  \@gls@warnontheglossdefined
  \renewenvironment{theglossary}{}{}%
}
```

The glossary header is given by `\glossaryheader`. This forms part of the glossary style, and must indicate what should appear immediately after the start of the `theglossary` environment. (For example, if the glossary uses a tabular-like environment, it may be used to set the header row.) Note that if you don't want a header row, the glossary style must redefine `\glossaryheader` to do nothing.

`\glossaryheader`

```
\newcommand*{\glossaryheader}{}
```

`\glstarget`

```
\glstarget{<label>}{<name>}
```

Provide user interface to `\@glstarget` to make it easier to modify the glossary style in the document.

```
\newcommand*{\glstarget}[2]{\@glstarget{\glslinkprefix#1}{#2}}
```

As from version 3.08, glossary information is now written to the external files using `\glossentry` and `\subglossentry` instead of `\glossaryentryfield` and `\glossarysubentryfield`. The default definition provides backward compatibility for glossary styles that use the old forms.

`\compatibleglossentry`

```
\compatibleglossentry{<label>}{<page-list>}
```

Removed in v4.50.

`\glossentry`

```
\glossentry{<label>}{<page-list>}
```

```
\newcommand{\glossentry}[2]{\glossentryname{#1} #2}
```

`\glossentryname`

```
\newcommand*{\glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
```

```

        \letcs{\glo@name}{glo@glstetoklabel{#1}@name}%
        \expandafter\glsnamefont\expandafter{\glo@name}%
    }%
}

```

`\Glossentryname`

```

\newcommand*{\Glossentryname}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
        \glsnamefont{\Glsentryname{#1}}%
    }%
}
\glsmfuaddmap{\glossentryname}{\Glossentryname}

```

`\glossentrydesc`

```

\newcommand*{\glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
        \glentrydesc{#1}%
    }%
}

```

`\Glossentrydesc`

```

\newcommand*{\Glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
        \Glsentrydesc{#1}%
    }%
}
\glsmfuaddmap{\glossentrydesc}{\Glossentrydesc}

```

`\glossentrysymbol`

```

\newcommand*{\glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
        \glentrysymbol{#1}%
    }%
}

```

`\Glossentrysymbol`

```

\newcommand*{\Glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
        \Glsentrysymbol{#1}%
    }%
}
\glsmfuaddmap{\glossentrysymbol}{\Glossentrysymbol}

```

`\compatiblesubglossentry`

```
\compatiblesubglossentry{<level>}{<label>}{<page-list>}
```

Removed in v4.50

`\setglossentrycompatibility`

`\glossaryentryfield`

```
\glossaryentryfield{<label>}{<name>}{<description>}  
{<symbol>}{<page-list>}
```

This command formerly governed how each entry row should be formatted in the glossary. Now deprecated. Removed in v4.50

`\glossarysubentryfield`

```
\glossarysubentryfield{<level>}{<label>}{<name>}  
{<description>}{<symbol>}{<page-list>}
```

This command governs how each subentry should be formatted in the glossary. Glossary styles need to redefine this command. Most of the predefined styles ignore `<symbol>`. The first argument is a number indicating the level. (The level should be greater than or equal to 1.)

Removed in v4.50.

`\subglossentry`

```
\subglossentry{<level>}{<label>}{<page-list>}
```

```
\newcommand{\subglossentry}[3]{\glossentryname{#2} #3}
```

Within each glossary, the entries form distinct groups which are determined by the first character of the sort key. When using `makeindex`, there will be a maximum of 28 groups: symbols, numbers, and the 26 alphabetical groups A, . . . , Z. If you use `xindy` the groups will depend on whatever alphabet is used. This is determined by the language or custom alphabets can be created in the `xindy` style file. The command `\glsgroupskip` specifies what to do between glossary groups. Glossary styles must redefine this command. (Note that `\glsgroupskip` only occurs between groups, not at the start or end of the glossary.)

`\glsgroupskip`

```
\newcommand*{\glsgroupskip}{} 
```

Each of the 28 glossary groups described above is preceded by a group heading. This is formatted by the command `\glsgroupheading` which takes one argument which is the *label* assigned to that group (not the title). The corresponding labels are: `glsymbols`, `glsnumbers`, A, . . . , Z. Glossary styles must

redefined this command. (In between groups, `\glsgroupheading` comes immediately after `\glsgroupskip`.)

`\glsgroupheading`

```
\newcommand*{\glsgroupheading}[1]{}
```

It is possible to “trick” `makeindex` into treating entries as though they belong to the same group, even if the terms don’t start with the same letter, by modifying the `sort key`. For example, all entries belonging to one group could be defined so that the `sort key` starts with an `a`, while entries belonging to another group could be defined so that the `sort key` starts with a `b`, and so on. If you want each group to have a heading, you would then need to modify the translation control sequences `\glsgetgrouptitle` and `\glsgetgrouplabel` so that the label is translated into the required title (and vice-versa).

```
\glsgetgrouptitle{<label>}
```

This command produces the title for the glossary group whose label is given by `<label>`. By default, the group labelled `glssymbols` produces `\glssymbolsgroupname`, the group labelled `glsnumbers` produces `\glsnumbersgroupname` and all the other groups simply produce their label. As mentioned above, the group labels are: `glssymbols`, `glsnumbers`, `A`, `...`, `Z`. If you want to redefine the group titles, you will need to redefine this command. Languages other than English may produce labels that are non-expandable, so we need to check for that otherwise it will create a “missing `\endcsname` inserted” error.

`\glsgetgrouptitle`

```
\newrobustcmd*{\glsgetgrouptitle}[1]{%
  \@gls@getgrouptitle{#1}{\@gls@grptitle}%
  \@gls@grptitle
}
```

`\@gls@getgrouptitle` Gets the group title specified by the label (first argument) and stores in the second argument, which must be a control sequence.

```
\ExplSyntaxOn
\newcommand*{\@gls@getgrouptitle}[2]{%
```

Note that `\dtl@ifsingle` (provided by `datatool`) is being phased out.

```
\def#2{#1}%
\regex_match:nnT { ^\A(.|glsnumbers|glssymbols)\Z } { #1 }
{ \ifcsundef{#1groupname}{ }\letcs#2{#1groupname} }
}
\ExplSyntaxOff
```

`\@gls@noidx@getgrouptitle` Version for the no-indexing app option:

```
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%
  \DTLifint{#1}%
  {\edef#2{\char#1\relax}}%
```

```

    {%
      \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
    }%
  }

```

```
\glsgetgrouplabel{<title>}
```

This command does the reverse to the previous command. The argument is the group title, and it produces the group label. Note that if you redefine `\glsgetgrouptitle`, you will also need to redefine `\glsgetgrouplabel`.

`\glsgetgrouplabel` This only works for the symbol and number groups, so is best avoided and not documented. However, it has been converted to use L^AT_EX3 commands.

```

\ExplSyntaxOn
\newcommand*{\glsgetgrouplabel}[1]{%
  \str_if_eq:eeTF {#1} { \glssymbolsgroupname }
  { \glssymbols }
  {
    \str_if_eq:eeTF {#1} { \glsnumbersgroupname }
    { \glsnumbers }
    { #1 }
  }
}
\ExplSyntaxOff

```

The command `\setentrycounter` sets the entry's associated counter (required by `\glshypernumber` etc.) `\glslink` and `\glsadd` encode the `\glossary` argument so that the relevant counter is set prior to the formatting command.

`\setentrycounter` The prefix is an optional argument because this command originally only had one argument. The optional argument was added in v3.0.

```

\newcommand*{\setentrycounter}[2] []{%
  \def\@glo@counterprefix{#1}%
  \ifx\@glo@counterprefix\@empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1}%
  \fi
  \def\glsentrycounter{#2}%
}

```

The current glossary style can be set using `\setglossarystyle{<style>}`.

`\setglossarystyle`

```

\newcommand*{\setglossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
}

```

```

    {%
      \csname @glsstyle@#1\endcsname
    }%
  Set the default style if it's not already set.
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}

```

`\glossarystyle` Deprecated in v3.08a. Removed in v4.50.

`\newglossarystyle` New glossary styles can be defined using:

```
\newglossarystyle{<name>}{<definition>}
```

The `<definition>` argument should redefine `\theglossary`, `\glossaryheader`, `\glsgroupheading`, `\glossaryentryfield` and `\glsgroupskip` (see [subsection 2.19](#) for the definitions of predefined styles). Glossary styles should not redefine `\glossarypreamble` and `\glossarypostamble`, as the user should be able to switch between styles without affecting the pre- and postambles.

```

\newcommand{\newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}

```

`\renewglossarystyle` Code for this macro supplied by Marco Daniel.

```

\newcommand{\renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}

```

Glossary entries are encoded so that the second argument to `\glossaryentryfield` is always specified as `\glsnamefont{<name>}`. This allows the user to change the font used to display the name term without having to redefine `\glossaryentryfield`. The default uses the surrounding font, so in the list type styles (which place the name in the optional argument to `\item`) the name will appear in bold.

`\glsnamefont`

```
\newcommand*{\glsnamefont}[1]{#1}
```

Each glossary entry has an associated number list (usually page numbers) that indicate where in the document the entry has been used. The format for these number lists can be changed using the `format` key in commands like `\glslink`. The default format is given by `\glsnumber`. This takes a single argument which may be a single number, a number range or a number list. The number ranges are delimited with `\delimR`, the number lists are delimited with `\delimN`.

If the document doesn't have hyperlinks, the numbers can be displayed just as they are, but if the document supports hyperlinks, the numbers should link to the relevant location. This means extracting the individual numbers from the list or ranges. The `hyperref` package does this with the `\hyperpage` command, but this is encoded for comma and dash delimiters and only for the page counter, but this code needs to be more general. So I have adapted the code used in the `hyperref` package.

`\glsnumber`

```
\ifcsundef{hyperlink}%  
{%  
  \def\glsnumber#1{#1}%  
}%  
{%  
  \def\glsnumber#1{\@xp@glshypernumber{#1}}  
}
```

`\@xp@glshypernumber`

```
\newrobustcmd{\@xp@glshypernumber}[1]{%  
  \@glsnumber#1\nohyperpage{ }\@nil  
}
```

`\@glsnumber` This code was provided by Heiko Oberdiek to allow material to be attached to the location.

```
\def\@glsnumber#1\nohyperpage#2#3\@nil{%  
  \ifx\#1\%  
    \else  
      \gls@delimR#1\delimR\delimR\%  
    \fi  
    \ifx\#2\%  
      \else  
        #2%  
      \fi  
    \ifx\#3\%  
      \else  
        \@glsnumber#3\@nil  
      \fi  
}
```

`\gls@delimR` displays a range of numbers for the counter whose name is given by `\@gls@counter` (which must be set prior to using `\gls@hypernumber`).

```
\gls@delimR
\def\gls@delimR#1\delimR #2\delimR #3\{\%
\ifx\#2\%
\gls@delimN{#1}%
\else
\@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
\fi
}
```

`\gls@delimN` displays a list of individual numbers, instead of a range:

```
\gls@delimN
\def\gls@delimN#1{\gls@delimN#1\delimN \delimN\}
\def\gls@delimN#1\delimN #2\delimN#3\{\%
\ifx\#3\%
\@gls@numberlink{#1}%
\else
\@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
\fi
}
```

The following code is modified from `hyperref`'s `\HyInd@pagelink` where the name of the counter being used is given by `\@gls@counter`.

```
\def\@gls@numberlink#1{%
\begingroup
\toks@={}%
\@gls@removespaces#1 \@nil
\endgroup
}

\def\@gls@removespaces#1 #2\@nil{%
\toks@=\expandafter{\the\toks@#1}%
\ifx\#2\%

\edef\@gls@x{\the\toks@}%
\ifx\@gls@x\empty
\else

\bgroup
```

Allow any markup to be stripped from the location when forming the target name.

```
\glswrglossdisableanchorcmds
\protected@edef\@gls@tmp{\gls@entrycounter\@gls@counterprefix
\glswrglosslocationtarget{\@gls@x}}%
```

If anything hasn't been fully expanded by this point then it will cause a problem, so sanitize to prevent an error, but it's likely the target won't be valid.


```

\@onelevel@sanitize\@gls@tmp
\edef\@gls@x{\noexpand\egroup
\noexpand\hyperlink{\@gls@tmp}%
{\noexpand\glswrglosslocationtextfmt{\the\toks@}}%
}%
\@gls@x
\fi
\else
\@gls@ReturnAfterFi{%
\@gls@removespaces#2\@nil
}%
\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}

```

The following commands will switch to the appropriate font, and create a hyperlink, if hyperlinks are supported. If hyperlinks are not supported, they will just display their argument in the appropriate font.

```

\hyperrm
\newcommand*\hyperrm[1]{\textrm{\glsnumber{#1}}}

\hypersf
\newcommand*\hypersf[1]{\textsf{\glsnumber{#1}}}

\hypertt
\newcommand*\hypertt[1]{\texttt{\glsnumber{#1}}}

\hyperbf
\newcommand*\hyperbf[1]{\textbf{\glsnumber{#1}}}

\hypermd
\newcommand*\hypermd[1]{\textmd{\glsnumber{#1}}}

\hyperit
\newcommand*\hyperit[1]{\textit{\glsnumber{#1}}}

\hypersl
\newcommand*\hypersl[1]{\textsl{\glsnumber{#1}}}

\hyperup
\newcommand*\hyperup[1]{\textup{\glsnumber{#1}}}

\hypersc
\newcommand*\hypersc[1]{\textsc{\glsnumber{#1}}}

\hyperemph
\newcommand*\hyperemph[1]{\emph{\glsnumber{#1}}}

```

2.17 Acronyms

`\oldacronym`

```
\oldacronym[⟨label⟩]{⟨abbrv⟩}{⟨long⟩}{⟨key-val list⟩}
```

This emulates the way the old `glossary` package defined acronyms. It is equivalent to `\newacronym[⟨key-val list⟩]{⟨label⟩}{⟨abbrv⟩}{⟨long⟩}` and it additionally defines the command `\⟨label⟩` which is equivalent to `\gls{⟨label⟩}` (thus `⟨label⟩` must only contain alphabetical characters). If `⟨label⟩` is omitted, `⟨abbrv⟩` is used. This only emulates the syntax of the old `glossary` package. The way the acronyms appear in the list of acronyms is determined by the definition of `\newacronym` and the glossary style.

Note that `\⟨label⟩` can't have an optional argument if the `xspace` package is loaded. If `xspace` hasn't been loaded then you can do `\⟨label⟩[⟨insert⟩]` but you can't do `\⟨label⟩[⟨key-val list⟩]`. For example if you define the acronym `svm`, then you can do `\svm['s]` but you can't do `\svm[format=textbf]`. If the `xspace` package is loaded, `\svm['s]` will appear as `svm ['s]` which is unlikely to be the desired result. In this case, you will need to use `\gls` explicitly, e.g. `\gls{svm}['s]`. Note that it is up to the user to load `xspace` if desired.

```
\newcommand{\oldacronym}[4][\gls@label]{%
  \def\gls@label#2{%
    \newacronym[#4]{#1}{#2}{#3}%
    \ifcsundef{xspace}%
      {%
        \expandafter\edef\csname#1\endcsname{%
          \noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}%
        }%
      }%
    }%
  }%
  \def
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
        \noexpand\gls{#1}\noexpand\xspace}%
    }%
  }%
}
```

```
\newacronym[⟨key-val list⟩]{⟨label⟩}{⟨abbrv⟩}{⟨long⟩}
```

This is a quick way of defining acronyms, using `\newglossaryentry` with the appropriate values. It sets the glossary type to `\acronymtype` which will be `acronym` if the package option `acronym` has been used, otherwise it will be the default glossary. Since `\newacronym` merely calls `\newglossaryentry`, the acronym is treated like any other glossary entry.

If you prefer a different format, you can redefine `\newacronym` as required. The optional argument can be used to override any of the settings.

This is just a stub. It's redefined by `\SetGenericNewAcronym` and by `glossaries-extra`. It used to also be redefined by the obsolete acronym styles

that have now been removed. If it is used then it means that the acronym style hasn't been initialised.

```
\newacronym
\newcommand{\newacronym}[4] []{%
\gls@newacronym{#1}{#2}{#3}{#4}%
}
```

`\gls@newacronym`

```
\newcommand{\gls@newacronym}[4]{%
```

Setting the acronym style will redefine `\newacronym` because it calls `\SetGenericNewAcronym` but if something weird has happened to alter this, redefine `\gls@newacronym` now to prevent an infinite loop.

```
\renewcommand{\gls@newacronym}[4]{%
\PackageError{glossaries}{incompatible acronym style:
rollback may be required}%
{something has gone wrong with \string\newacronym. Try
\string\usepackage{glossaries}[=v4.49]}%
}%
\setacronymstyle{long-short}%
\newacronym[#1]{#2}{#3}{#4}%
}
```

Set up some convenient short cuts. These need to be changed if `\newacronym` is changed (or if the description key is changed).

`\acrpluralsuffix` Plural suffix used by `\newacronym`. This just defaults to `\glspluralsuffix` but is changed to include `\textup` if the smallcaps option is used, so that the suffix doesn't appear in small caps as it doesn't look right. For example, ABCS looks as though the "s" is part of the acronym, but ABCs looks as though the "s" is a plural suffix. Since the entire text `abcs` is set in `\textsc`, `\textup` is need to cancel it out.

```
\newcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}
```

If `garamondx` has been loaded, need to use `\textulc` instead of `\textup`.

`\glstextup`

```
\newrobustcmd*{\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
```

The following are defined for compatibility with version 2.07 and earlier.

`\glsshortkey`

```
\newcommand*{\glsshortkey}{short}
```

`\glsshortpluralkey`

```
\newcommand*{\glsshortpluralkey}{shortplural}
```

`\glslongkey`

```
\newcommand*{\glslongkey}{long}
```

`\glslongpluralkey`

```
\newcommand*\glslongpluralkey}{longplural}
```

Using `\acrlinkfullformat` and `\acrfullformat` is now deprecated. However, these commands are still used in the initial definition of `\acrfull` and `\glsentryfull`. The acronym style now redefines `\glsentryfull`, and `\acrfull` now uses `\acrfullfmt`, which is redefined by the acronym style. Similar for the case and plural variants. This means that `\acrlinkfullformat` and `\acrfullformat` should now longer be used with v4.50, which will ensure that `\setacronymstyle` is used on the first instance of `\newacronym` if a style hasn't been set.

Outlier cases may occur where someone has defined an entry explicitly with `\newglossaryentry` and set the long and short keys and is using `\glsentryfull` or `\acrfull` without a style set. So, for now, the deprecated `\acrlinkfullformat` and `\acrfullformat` are retained even though the remaining obsolete acronym code has been removed. However, they may be removed in a future version.

`\acrlinkfullformat` Format for full links like `\acrfull`. Syntax: `\acrlinkfullformat{<long cs>}{<short cs>}{<options>}{<label>}{<insert>}` Deprecated.

```
\newcommand{\acrlinkfullformat}[5]{%
  \acrfullformat{#1{#3}{#4}[#5]}{#2{#3}{#4}[]}%
}
```

`\acrfullformat` Default full form is `<long>` (`<short>`). Deprecated.

```
\newcommand{\acrfullformat}[2]{#1\glsspace{#2}}
```

`\glsspace` Robust space to ensure it's written to the `.glsdefs` file.

```
\newrobustcmd{\glsspace}{\space}
```

The following `\acrfullfmt` set of commands are used by the corresponding `\acrfull` command, and are redefined by the acronym style to match the style's full form.

`\acrfullfmt` No case change full format. This default will be overridden by the acronym style, so this definition shouldn't actually be used any more, but it still needs to be defined so the it doesn't cause a problem when acronym styles redefine it. This is initialised to use the deprecated `\acrlinkfullformat` but will be redefined by the acronym style when it's set.

```
\newcommand*\acrfullfmt}[3]{%
  \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
```

`\Acrfullfmt` Sentence case full format.

```
\newcommand*\Acrfullfmt}[3]{%
  \acrlinkfullformat{\@Acrlong}{\@Acrshort}{#1}{#2}{#3}%
}
```

```

\ACRfullfmt All upper case full format.
  \newcommand*\ACRfullfmt}[3]{%
    \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
  }

\acrfullplfmt No case change plural full format.
  \newcommand*\acrfullplfmt}[3]{%
    \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
  }

\Acrfullplfmt Sentence case plural full format.
  \newcommand*\Acrfullplfmt}[3]{%
    \acrlinkfullformat{\@Acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
  }

\ACRfullplfmt All upper case plural full format.
  \newcommand*\ACRfullplfmt}[3]{%
    \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
  }

\acrfull Full form of the acronym.
  \newrobustcmd*\acrfull{\@gls@hyp@opt\ns@acrfull}

  \newcommand*\ns@acrfull[2][ ]{%
    \new@ifnextchar[{\@acrfull{#1}{#2}}%
      {\@acrfull{#1}{#2}[ ]}%
  }

\@acrfull Low-level macro:
  \def\@acrfull#1#2[#3]{%
    Make it easier for acronym styles to change this:
    \acrfullfmt{#1}{#2}{#3}%
  }

  Default format for full acronym

\Acrfull
  \newrobustcmd*\Acrfull{\@gls@hyp@opt\ns@Acrfull}
  Add sentence case mapping.
  \glsmfuaddmap{\acrfull}{\Acrfull}

  \newcommand*\ns@Acrfull[2][ ]{%
    \new@ifnextchar[{\@Acrfull{#1}{#2}}%
      {\@Acrfull{#1}{#2}[ ]}%
  }

  Low-level macro:
  \def\@Acrfull#1#2[#3]{%

```

Make it easier for acronym styles to change this:

```
\acrfulfmt{#1}{#2}{#3}%  
}
```

`\ACRfull`

```
\newrobustcmd*{\ACRfull}{\@gls@hyp@opt\ns@ACRfull}  
  
\newcommand*\ns@ACRfull[2] []{%  
  \new@ifnextchar[{\@ACRfull{#1}{#2}}%  
    {\@ACRfull{#1}{#2} []}%  
}
```

Low-level macro:

```
\def\@ACRfull#1#2[#3]{%
```

Make it easier for acronym styles to change this:

```
\acrfulfmt{#1}{#2}{#3}%  
}
```

Plural:

`\acrfullpl`

```
\newrobustcmd*{\acrfullpl}{\@gls@hyp@opt\ns@acrfullpl}  
  
\newcommand*\ns@acrfullpl[2] []{%  
  \new@ifnextchar[{\@acrfullpl{#1}{#2}}%  
    {\@acrfullpl{#1}{#2} []}%  
}
```

Low-level macro:

```
\def\@acrfullpl#1#2[#3]{%
```

Make it easier for acronym styles to change this:

```
\acrfullplfmt{#1}{#2}{#3}%  
}
```

`\Acrfullpl`

```
\newrobustcmd*{\Acrfullpl}{\@gls@hyp@opt\ns@Acrfullpl}
```

Add sentence case mapping.

```
\glsmfuaddmap{\acrfullpl}{\Acrfullpl}  
  
\newcommand*\ns@Acrfullpl[2] []{%  
  \new@ifnextchar[{\@Acrfullpl{#1}{#2}}%  
    {\@Acrfullpl{#1}{#2} []}%  
}
```

Low-level macro:

```
\def\@Acrfullpl#1#2[#3]{%
```

Make it easier for acronym styles to change this:

```
\Acrfullplfmt{#1}{#2}{#3}%  
}
```

```

\ACRfullpl
\newrobustcmd*\ACRfullpl{\@gls@hyp@opt\ns@ACRfullpl}

\newcommand*\ns@ACRfullpl[2][\%
  \new@ifnextchar[{\@ACRfullpl{#1}{#2}}%
    {\@ACRfullpl{#1}{#2}[]}%
  ]

```

Low-level macro:

```

\def\@ACRfullpl#1#2[#3]{%
  \ACRfullplfmt{#1}{#2}{#3}%
}

```

Make it easier for acronym styles to change this:

2.18 Predefined acronym styles

`\acronymfont` Used by the acronym styles to apply a font change to the short form.

```
\newcommand{\acronymfont}[1]{#1}
```

`\firstacronymfont` Used by the acronym styles to apply a font change to the short form on first use.

```
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
```

`\acrnameformat` The styles that allow an additional description use `\acrnameformat{<short>}{<long>}` to determine what information is displayed in the name.

```
\newcommand*\acrnameformat[2]{\acronymfont{#1}}
```

Define some tokens used by `\newacronym`:

`\glskeylisttok`

```
\newtoks\glskeylisttok
```

`\glslabeltok`

```
\newtoks\glslabeltok
```

`\glsshorttok`

```
\newtoks\glsshorttok
```

`\gslongtok`

```
\newtoks\gslongtok
```

`\newacronymhook` Provide a hook for `\newacronym`:

```
\newcommand*\newacronymhook{}
```

`\SetGenericNewAcronym` New improved version of setting the acronym style.

```
\newcommand*\SetGenericNewAcronym{%

```

Change the behaviour of \Glsentryname to workaround expansion issues that cause a problem for \makefirstuc

```
\let\@Gls@entryname\@Gls@acentryname
```

Change the way acronyms are defined:

```
\renewcommand{\newacronym}[4][]{%
  \ifdefempty{\@glsacronymlists}%
  {%
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
  }%
  {}%
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \glslongtok{##4}%
  \newacronymhook
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\expandonce{\acronymentry{##2}}},%
      sort={\acronymsort{\the\glsshorttok}{\the\glslongtok}},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%

      longplural={\the\glslongtok\noexpand\glspluralsuffix},%
      \GenericAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}%
```

Make sure that \acrfull etc reflects the new style:

```
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsuppercase{\genacrfullformat{##2}{##3}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
```



```

\glslink[##1]{##2}{%
\glsuppercase{\genplacrfullformat{##2}{##3}}}%

```

Make sure that `\glsentryfull` etc reflects the new style:

```

\renewcommand*\glsentryfull}[1]{\genacrfullformat{##1}{}}%
\renewcommand*\Glsentryfull}[1]{\Genacrfullformat{##1}{}}%
\renewcommand*\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}%
\renewcommand*\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}%
}

```

`\GenericAcronymFields` Fields used by `\SetGenericNewAcronym` that can be changed by the acronym style.

```

\newcommand*\GenericAcronymFields{description={\the\glslongtok}}

```

`\acronymentry`

```

\acronymentry{<label>}

```

Display style for the name field in the list of acronyms.

```

\newcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{#1}}}

```

`\acronymsort`

```

\acronymsort{<short>}{<long>}

```

Default sort format for acronyms.

```

\newcommand*\acronymsort}[2]{#1}

```

`\setacronymstyle`

```

\setacronymstyle{<style name>}

```

```

\newcommand*\setacronymstyle}[1]{%
\ifcsundef{@glsacr@dispstyle@#1}
{%
\PackageError{glossaries}{Undefined acronym style ‘#1’}{%
}%
}%
{%
\ifdefempty{\@glsacronymlists}%
{%
\DeclareAcronymList{\acronymtype}%
}%
{}%
\SetGenericNewAcronym
\GlsUseAcrStyleDefs{#1}%
\@for\@gls@type:=\@glsacronymlists\do{%
\defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{#1}}%
}%
}%
}

```

```
\newacronymstyle{<style name>}{<entry format
definition>}{<display definitions>}
```

`\newacronymstyle`

Defines a new acronym style called *<style name>*.

```
\newcommand*{\newacronymstyle}[3]{%
  \ifcsdef{@glsacr@dispstyle@#1}%
  {%
    \PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
  }%
  {%
    \csdef{@glsacr@dispstyle@#1}{#2}%
    \csdef{@glsacr@styledefs@#1}{#3}%
  }%
}
```

`\renewacronymstyle` Redefines the given acronym style.

```
\newcommand*{\renewacronymstyle}[3]{%
  \ifcsdef{@glsacr@dispstyle@#1}%
  {%
    \csdef{@glsacr@dispstyle@#1}{#2}%
    \csdef{@glsacr@styledefs@#1}{#3}%
  }%
  {%
    \PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
  }%
}
```

`\GlsUseAcrEntryDispStyle`

```
\newcommand*{\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
```

`\GlsUseAcrStyleDefs`

```
\newcommand*{\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
```

Predefined acronym styles:

`long-short` *<long>* (*<short>*) acronym style.

```
\newacronymstyle{long-short}%
{%
```

Check for long form in case this is a mixed glossary.

```
\ifglshaslong{\glslabel}{\glsacrfmt}{\glsacrfmt}%
}%
{%
```

```
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
```

```
\renewcommand*{\genacrfullformat}[2]{%
```

```
\glsentrylong{##1}##2\space
```

```
(\protect\firstacronymfont{\glsentryshort{##1}})%
```

```
}%
```

```
\renewcommand*{\Genacrfullformat}[2]{%
```

```

\Glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*\genplacrfullformat}[2]{%
\glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix*\glspluralsuffix}%
}

```

long-sp-short Similar to the previous style but allows the space between the long and short form to be customized.

```

\newacronymstyle{long-sp-short}%
{%

```

Check for long form in case this is a mixed glossary.

```

\ifglshaslong{\glslabel}{\glsacspace}{\glsacspace}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat}[2]{%
\glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*\genplacrfullformat}[2]{%
\glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix*\glspluralsuffix}%
}

```

`\glsacspace` Space between long and short form for the above style. This uses a non-breakable space if the short form is less than 3em, otherwise it uses a regular space. It's unlikely that the short form will contain any instances of `\gls` so this just uses `\settowidth` instead of `\glsmeasurewidth`.

```
\newcommand*\glsacspace}[1]{%
  \settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{##1}})}%
  \ifdim\dimen@<3em~\else\space\fi
}
```

`short-long` $\langle short \rangle$ ($\langle long \rangle$) acronym style.

```
\newacronymstyle{short-long}%
{%
```

Check for long form in case this is a mixed glossary.

```
\ifglsashaslong{\glslabel}{\glsacacfmt}{\glsacacentryfmt}%
}%
{%
  \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
  \renewcommand*\genacrfullformat}[2]{%
    \protect\firstacronymfont{\glsentryshort{##1}}##2\space
    (\glsentrylong{##1})%
  }%
  \renewcommand*\Genacrfullformat}[2]{%
    \protect\firstacronymfont{\Glsentryshort{##1}}##2\space
    (\glsentrylong{##1})%
  }%
  \renewcommand*\genplacrfullformat}[2]{%
    \protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
    (\glsentrylongpl{##1})%
  }%
  \renewcommand*\Genplacrfullformat}[2]{%
    \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
    (\glsentrylongpl{##1})%
  }%

  \renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
  \renewcommand*\acronymssort}[2]{##1}%
  \renewcommand*\acronymfont}[1]{##1}%
  \renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
  \renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
```

`long-sc-short` $\langle long \rangle$ ($\langle \textsc{short} \rangle$) acronym style.

```
\newacronymstyle{long-sc-short}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
```

```

        \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
    }

long-sm-short <long> (\textsmaller{<short>}) acronym style.
    \newacronymstyle{long-sm-short}%
    {%
        \GlsUseAcrEntryDispStyle{long-short}%
    }%
    {%
        \GlsUseAcrStyleDefs{long-short}%
        \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
        \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
    }

sc-short-long <short> (\textsc{<long>}) acronym style.
    \newacronymstyle{sc-short-long}%
    {%
        \GlsUseAcrEntryDispStyle{short-long}%
    }%
    {%
        \GlsUseAcrStyleDefs{short-long}%
        \renewcommand{\acronymfont}[1]{\textsc{##1}}%
        \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
    }

sm-short-long <short> (\textsmaller{<long>}) acronym style.
    \newacronymstyle{sm-short-long}%
    {%
        \GlsUseAcrEntryDispStyle{short-long}%
    }%
    {%
        \GlsUseAcrStyleDefs{short-long}%
        \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
        \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
    }

long-short-desc <long> ({<short>}) acronym style that has an accompanying description (which
the user needs to supply).
    \newacronymstyle{long-short-desc}%
    {%
        \GlsUseAcrEntryDispStyle{long-short}%
    }%
    {%
        \GlsUseAcrStyleDefs{long-short}%
        \renewcommand*{\GenericAcronymFields}{}%
        \renewcommand*{\acronymsort}[2]{##2}%
        \renewcommand*{\acronymentry}[1]{%
            \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
    }

```

`long-sp-short-desc` $\langle long \rangle$ ($\{ \langle short \rangle \}$) acronym style that has an accompanying description (which the user needs to supply). The space between the long and short form is given by `\glsacspace`.

```

\newacronymstyle{long-sp-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sp-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sp-short}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%
\glsentrylong{##1}\glsacspace{##1}(\acronymfont{\glsentryshort{##1}})}%
}

```

`long-sc-short-desc` $\langle long \rangle$ ($\backslash\text{textsc}\{ \langle short \rangle \}$) acronym style that has an accompanying description (which the user needs to supply).

```

\newacronymstyle{long-sc-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sc-short}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}

```

`long-sm-short-desc` $\langle long \rangle$ ($\backslash\text{textsmaller}\{ \langle short \rangle \}$) acronym style that has an accompanying description (which the user needs to supply).

```

\newacronymstyle{long-sm-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sm-short}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}

```

`short-long-desc` $\langle short \rangle$ ($\{ \langle long \rangle \}$) acronym style that has an accompanying description (which the user needs to supply).

```

\newacronymstyle{short-long-desc}%
{%
\GlsUseAcrEntryDispStyle{short-long}%
}%

```

```

    {%
      \GlsUseAcrStyleDefs{short-long}%
      \renewcommand*\GenericAcronymFields{}%
      \renewcommand*\acronymsort}[2]{##2}%
      \renewcommand*\acronymentry}[1]{%
        \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
    }
sc-short-long-desc <long> (\textsc{<short>}) acronym style that has an accompanying description
                    (which the user needs to supply).
                    \newacronymstyle{sc-short-long-desc}%
                    {%
                      \GlsUseAcrEntryDispStyle{sc-short-long}%
                    }%
                    {%
                      \GlsUseAcrStyleDefs{sc-short-long}%
                      \renewcommand*\GenericAcronymFields{}%
                      \renewcommand*\acronymsort}[2]{##2}%
                      \renewcommand*\acronymentry}[1]{%
                        \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
                    }
sm-short-long-desc <long> (\textsmaller{<short>}) acronym style that has an accompanying de-
                    scription (which the user needs to supply).
                    \newacronymstyle{sm-short-long-desc}%
                    {%
                      \GlsUseAcrEntryDispStyle{sm-short-long}%
                    }%
                    {%
                      \GlsUseAcrStyleDefs{sm-short-long}%
                      \renewcommand*\GenericAcronymFields{}%
                      \renewcommand*\acronymsort}[2]{##2}%
                      \renewcommand*\acronymentry}[1]{%
                        \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
                    }
dua <long> only acronym style.
    \newacronymstyle{dua}%
    {%
      Check for long form in case this is a mixed glossary.
      \ifdefempty\glscustomtext
      {%
        \ifglshaslong{\glslabel}%
        {%
          \glsifplural
          {%
            Plural form:
            \glscapscase
            {%

```

Plural form, don't adjust case:

```
\glsentrylongpl{\glslabel}\glsinsert  
}%  
{%
```

Plural form, make first letter upper case:

```
\Glsentrylongpl{\glslabel}\glsinsert  
}%  
{%
```

Plural form, all caps:

```
\glsuppercase  
{\glsentrylongpl{\glslabel}\glsinsert}%  
}%  
{%
```

Singular form

```
\glscapscase  
{%
```

Singular form, don't adjust case:

```
\glsentrylong{\glslabel}\glsinsert  
}%  
{%
```

Subsequent singular form, make first letter upper case:

```
\Glsentrylong{\glslabel}\glsinsert  
}%  
{%
```

Subsequent singular form, all caps:

```
\glsuppercase  
{\glsentrylong{\glslabel}\glsinsert}%  
}%  
}%  
{%
```

Not an acronym:

```
\glsgenentryfmt  
}%  
}%  
{\glscustomtext\glsinsert}%  
}%  
{%  
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%  
  
\renewcommand*{\acrfullfmt}[3]{%  
  \glslink[##1]{##2}{\glsentrylong{##2}##3\space  
  (\acronymfont{\glsentryshort{##2}})}%  
\renewcommand*{\Acrfullfmt}[3]{%  
  \glslink[##1]{##2}{\Glsentrylong{##2}##3\space
```



```

        (\acronymfont{\glsentryshort{##2}})}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsuppercase{\glsentrylong{##2}##3\space
      (\acronymfont{\glsentryshort{##2}})}}}%

\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\glsentrylongpl{##2}##3\space
    (\acronymfont{\glsentryshortpl{##2}})}}%

\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Glsentrylongpl{##2}##3\space
    (\acronymfont{\glsentryshortpl{##2}})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsuppercase{\glsentrylongpl{##2}##3\space
      (\acronymfont{\glsentryshortpl{##2}})}}}%
\renewcommand*{\glsentryfull}[1]{%
  \glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
  \glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}

```

dua-desc *<long>* only acronym style with user-supplied description.

```

\newacronymstyle{dua-desc}%
{%
  \GlsUseAcrEntryDispStyle{dua}%
}%
{%
  \GlsUseAcrStyleDefs{dua}%
\renewcommand*{\GenericAcronymFields}{}%

\renewcommand*{\acronymentry}[1]{\glsentrylong{##1}}%
\renewcommand*{\acronymsort}[2]{##2}%
}%

```

footnote *<short>*\footnote{*<long>*} acronym style.

```

\newacronymstyle{footnote}%
{%

```

Check for long form in case this is a mixed glossary.

```
\ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%  
}%  
{%  
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
```

Need to ensure hyperlinks are switched off on first use:

```
\glshyperfirstfalse  
\renewcommand*{\genacrfullformat}[2]{%  
\protect\firstacronymfont{\glsentryshort{##1}}##2%  
\protect\footnote{\glsentrylong{##1}}%  
}%  
\renewcommand*{\Genacrfullformat}[2]{%  
\firstacronymfont{\Glsentryshort{##1}}##2%  
\protect\footnote{\glsentrylong{##1}}%  
}%  
\renewcommand*{\genplacrfullformat}[2]{%  
\protect\firstacronymfont{\glsentryshortpl{##1}}##2%  
\protect\footnote{\glsentrylongpl{##1}}%  
}%  
\renewcommand*{\Genplacrfullformat}[2]{%  
\protect\firstacronymfont{\Glsentryshortpl{##1}}##2%  
\protect\footnote{\glsentrylongpl{##1}}%  
}%  
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%  
\renewcommand*{\acronymsort}[2]{##1}%  
\renewcommand*{\acronymfont}[1]{##1}%  
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
```

Don't use footnotes for \acrfull:

```
\renewcommand*{\acrfullfmt}[3]{%  
\glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space  
(\glsentrylong{##2})}%  
\renewcommand*{\Acrfullfmt}[3]{%  
\glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space  
(\glsentrylong{##2})}%  
\renewcommand*{\ACRfullfmt}[3]{%  
\glslink[##1]{##2}{%  
\glsupercase{\acronymfont{\glsentryshort{##2}}##3\space  
(\glsentrylong{##2})}}}%  
\renewcommand*{\acrfullplfmt}[3]{%  
\glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space  
(\glsentrylongpl{##2})}%  
\renewcommand*{\Acrfullplfmt}[3]{%  
\glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space  
(\glsentrylongpl{##2})}%  
\renewcommand*{\ACRfullplfmt}[3]{%  
\glslink[##1]{##2}{%  
\glsupercase{\acronymfont{\glsentryshortpl{##2}}##3\space  
(\glsentrylongpl{##2})}}}%
```

Similarly for `\glentryfull` etc:

```

\renewcommand*\glentryfull}[1]{%
  \acronymfont{\glentryshort{##1}}\space(\glentrylong{##1})}%
\renewcommand*\Glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\glentrylong{##1})}%
\renewcommand*\glentryfullpl}[1]{%
  \acronymfont{\glentryshortpl{##1}}\space(\glentrylongpl{##1})}%
\renewcommand*\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\glentrylongpl{##1})}%
}

```

`footnote-sc` `\textsc{⟨short⟩}\footnote{⟨long⟩}` acronym style.

```

\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
}%

```

`footnote-sm` `\textsmaller{⟨short⟩}\footnote{⟨long⟩}` acronym style.

```

\newacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*\acrpluralsuffix{\glsacrpluralsuffix}%
}%

```

`footnote-desc` `⟨short⟩\footnote{⟨long⟩}` acronym style that has an accompanying description (which the user needs to supply).

```

\newacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*\GenericAcronymFields{ }%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})%
  }
}

```

footnote-sc-desc \textsc{*short*}\footnote{*long*} acronym style that has an accompanying description (which the user needs to supply).

```
\newacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
```

footnote-sm-desc \textsmaller{*short*}\footnote{*long*} acronym style that has an accompanying description (which the user needs to supply).

```
\newacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
```

\DefineAcronymSynonyms

```
\newcommand*\DefineAcronymSynonyms}{%
```

Short form

```
\acs
```

```
\let\acs\acrshort
```

First letter uppercase short form

```
\Acs
```

```
\let\Acs\Acrshort
```

Plural short form

```
\acsp
```

```
\let\acsp\acrshortpl
```

First letter uppercase plural short form

```
\Acsp
```

```
\let\Acsp\Acrshortpl
```

Long form

`\acl`
`\let\acl\acrlong`

Plural long form

`\aclp`
`\let\aclp\acrlongpl`

First letter upper case long form

`\Acl`
`\let\Acl\Acrlong`

First letter upper case plural long form

`\Aclp`
`\let\Aclp\Acrlongpl`

Full form

`\acf`
`\let\acf\acrfull`

Plural full form

`\acfp`
`\let\acfp\acrfullpl`

First letter upper case full form

`\Acf`
`\let\Acf\Acrcfull`

First letter upper case plural full form

`\Acfp`
`\let\Acfp\Acrcfullpl`

Standard form

`\ac`
`\let\ac\gls`

First upper case standard form

`\Ac`
`\let\Ac\Gls`

Standard plural form

`\acp`
`\let\acp\glspl`

Standard first letter upper case plural form

`\Acp`

```
\let\Acp\Glspl
```

```
}
```

Define synonyms if required

```
\ifglsacrshortcuts
```

```
\DefineAcronymSynonyms
```

```
\fi
```

`\SetDefaultAcronymDisplayStyle` Sets the default acronym display style for given glossary. Used by `glossaries-extra` when restoring `\newacronym`.

```
\newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
```

```
\defglsentryfmt[#1]{\glsentryfmt}%
```

```
}
```

The following commands for setting the acronym style are now deprecated.

`\DefaultNewAcronymDef` Sets up the acronym definition for the default style. The information is provided by the tokens `\glslabeltok`, `\glsshorttok`, `\glslongtok` and `\glskeylisttok`. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\noexpand@assign@default@plurals` `\@gls@noexpand@assign@default@plurals` was used in `\DefaultNewAcronymDef`. Added in v4.47 to fix bug #162 (<https://www.dickimaw-books.com/bugtracker.php?key=162>). Removed in v4.50 along with the deprecated styles.

`\@gls@default@acr@plural`

`\gls@default@acr@firstplural`

`\gls@default@acr@descplural`

`\SetDefaultAcronymStyle` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\acrfootnote` This was used by the old footnote acronym styles. Since it may have been used outside of those styles, it has been retained.

```
\newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
```

`\acrlinkfootnote` This was used by the old footnote acronym styles. Since it may have been used outside of those styles, it has been retained.

```
\newcommand*{\acrlinkfootnote}[3]{%
```

```
\footnote{\glslink[#1]{#2}{#3}}%
```

```
}
```

<code>\acrlinkfootnote</code>	This was used by the old footnote acronym styles. Since it may have been used outside of those styles, it has been retained. <pre> \newcommand*{\acrlinkfootnote}[3]{% \footnote{#3}% } </pre>
<code>FootnoteAcronymDisplayStyle</code>	Sets the acronym display style for given glossary for the description and footnote combination. Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>DescriptionFootnoteNewAcronymDef</code>	Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>DescriptionFootnoteAcronymStyle</code>	If a description and footnote are both required, store the long form in the <code>symbol</code> key. Store the short form in <code>text</code> key. Note that since the long form is stored in the <code>symbol</code> key, if you want the long form to appear in the list of acronyms, you need to use a glossary style that displays the <code>symbol</code> key. This style was deprecated with the introduction of <code>\newacronymstyle</code> . Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>DescriptionDUAAcronymDisplayStyle</code>	Sets the acronym display style for given glossary with description and dua combination. Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>DescriptionDUANewAcronymDef</code>	Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>SetDescriptionDUAAcronymStyle</code>	Description, don't use acronym and no footnote. Note that the short form is stored in the <code>symbol</code> key, so if the short form needs to be displayed in the glossary, use a style the displays the <code>symbol</code> . This style was deprecated with the introduction of <code>\newacronymstyle</code> . Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>DescriptionAcronymDisplayStyle</code>	Sets the acronym display style for given glossary using the description setting (but not footnote or dua). Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\DescriptionNewAcronymDef</code>	Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\SetDescriptionAcronymStyle</code>	Option description is used, but not dua or footnote. Store long form in first key and short form in <code>text</code> and <code>symbol</code> key. The name is stored using <code>\acrnameformat</code> to allow the user to override the way the name is displayed in the list of acronyms. Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>FootnoteAcronymDisplayStyle</code>	Sets the acronym display style for given glossary with footnote setting (but not description or dua). Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.

`\FootnoteNewAcronymDef` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\SetFootnoteAcronymStyle` If footnote package option is specified, set the first use to append the long form (stored in `description`) as a footnote. Use the `description` key to store the long form. This style was deprecated with the introduction of `\newacronymstyle`.
 Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\glsdoparenifnotempty` Do a space followed by the argument if the argument doesn't expand to empty or `\relax`. If argument isn't empty (or `\relax`), apply the macro to it given in the second argument.

```

\newrobustcmd*{\glsdoparenifnotempty}[2]{%
  \protected@edef\gls@tmp{#1}%
  \ifdefempty\gls@tmp
  {}%
  {%
    \ifx\gls@tmp\@gls@default@value
    \else
    \space (#2{#1})%
    \fi
  }%
}

```

`\SetSmallAcronymDisplayStyle` Sets the acronym display style for given glossary where neither footnote nor description is required, but smallcaps or smaller specified. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\SmallNewAcronymDef` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\SetSmallAcronymStyle` Neither footnote nor description required, but smallcaps or smaller specified. Use the `symbol` key to store the short form and `first` to store the long form. This style was deprecated with the introduction of `\newacronymstyle`. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\SetDUADisplayStyle` Sets the acronym display style for given glossary with `dua` setting. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\DUANewAcronymDef` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\SetDUAStyle` Always expand acronyms. This style was deprecated with the introduction of `\newacronymstyle`. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\@gls@deprecated@acr@warning`

```

\newcommand*{\@gls@deprecated@acr@warning}[3]{%
  \GlossariesWarning{Acronym style option #1 deprecated.

```



```

    Use \string\setacronymstyle{#2} with just glossaries.sty or
    use \string\setabbreviationstyle[acronym]{#3} with
    glossaries-extra.sty}%
}

```

`\SetAcronymStyle` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

Allow user to define their own custom acronyms. (For compatibility with versions before v3.0, the short form is stored in the user1 key, the plural short form is stored in the user2 key, the long form is stored in the user3 key and the plural long form is stored in the user4 key.) Defaults to displaying only the acronym with the long form as the description.

`\SetCustomDisplayStyle` Sets the acronym display style. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\CustomAcronymFields` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\CustomNewAcronymDef` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\SetCustomStyle` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

2.19 Predefined Glossary Styles

The glossaries bundle comes with some predefined glossary styles. These need to be loaded now for the style option to use them.

First, the glossary hyper-navigation commands need to be loaded.

```
\RequirePackage{glossary-hypernav}
```

The styles that use list-like environments. These are not loaded if the `nolist` option is used:

```
\@gls@loadlist
```

The styles that use the `longtable` environment. These are not loaded if the `nolong` package option is used.

```
\@gls@loadlong
```

The styles that use the `supertabular` environment. These are not loaded if the `nosuper` package option is used or if the `supertabular` package isn't installed.

```
\@gls@loadsuper
```

The tree-like styles. These are not loaded if the `notree` package option is used.

```
\@gls@loadtree
```

The default glossary style is set according to the `style` package option, but can be overridden by `\glossarystyle`. The required style must be defined at this point.

```
\ifx\@glossary@default@style\relax
\else
\setglossarystyle{\@glossary@default@style}
\fi
```

2.20 Debugging Commands

`\showgloparent`

```
\showgloparent{<label>}
```

```
\newcommand*\showgloparent[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
}
```

`\showglolevel`

```
\showglolevel{<label>}
```

```
\newcommand*\showglolevel[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
}
```

`\showglotext`

```
\showglotext{<label>}
```

```
\newcommand*\showglotext[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
}
```

`\showgloplural`

```
\showgloplural{<label>}
```

```
\newcommand*\showgloplural[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@plural\endcsname
}
```

`\showglofirst`

```
\showglofirst{<label>}
```

```
\newcommand*\showglofirst}[1]{%
  \expandafter\show\csname glo\glsdetoklabel{#1}@first\endcsname
}
```

\showglofirstpl

```
\showglofirstpl{\label}
```

```
\newcommand*\showglofirstpl}[1]{%
  \expandafter\show\csname glo\glsdetoklabel{#1}@firstpl\endcsname
}
```

\showgloftype

```
\showgloftype{\label}
```

```
\newcommand*\showgloftype}[1]{%
  \expandafter\show\csname glo\glsdetoklabel{#1}@ftype\endcsname
}
```

\showglocounter

```
\showglocounter{\label}
```

```
\newcommand*\showglocounter}[1]{%
  \expandafter\show\csname glo\glsdetoklabel{#1}@counter\endcsname
}
```

\showglouseri

```
\showglouseri{\label}
```

```
\newcommand*\showglouseri}[1]{%
  \expandafter\show\csname glo\glsdetoklabel{#1}@useri\endcsname
}
```

\showglouserii

```
\showglouserii{\label}
```

```
\newcommand*\showglouserii}[1]{%
  \expandafter\show\csname glo\glsdetoklabel{#1}@userii\endcsname
}
```

`\showglouseriii` `\showglouseriii{<label>}`

```
\newcommand*{\showglouseriii}[1]{%  
  \expandafter\show\csname glo@glstdetoklabel{#1}@useriii\endcsname  
}
```

`\showglouseriv` `\showglouseriv{<label>}`

```
\newcommand*{\showglouseriv}[1]{%  
  \expandafter\show\csname glo@glstdetoklabel{#1}@useriv\endcsname  
}
```

`\showglouserv` `\showglouserv{<label>}`

```
\newcommand*{\showglouserv}[1]{%  
  \expandafter\show\csname glo@glstdetoklabel{#1}@userv\endcsname  
}
```

`\showglouservi` `\showglouservi{<label>}`

```
\newcommand*{\showglouservi}[1]{%  
  \expandafter\show\csname glo@glstdetoklabel{#1}@uservi\endcsname  
}
```

`\showgloname` `\showgloname{<label>}`

```
\newcommand*{\showgloname}[1]{%  
  \expandafter\show\csname glo@glstdetoklabel{#1}@name\endcsname  
}
```

`\showglodesc` `\showglodesc{<label>}`

```
\newcommand*{\showglodesc}[1]{%  
  \expandafter\show\csname glo@glstdetoklabel{#1}@desc\endcsname  
}
```

`\showgloDESCplural`

```
\showgloDESCplural{<label>}
```

```
\newcommand*{\showgloDESCplural}[1]{%  
  \expandafter\show\csname glo@glSDetoklabel{#1}@DESCplural\endcsname  
}
```

`\showgloSort`

```
\showgloSort{<label>}
```

```
\newcommand*{\showgloSort}[1]{%  
  \expandafter\show\csname glo@glSDetoklabel{#1}@Sort\endcsname  
}
```

`\showgloSymbol`

```
\showgloSymbol{<label>}
```

```
\newcommand*{\showgloSymbol}[1]{%  
  \expandafter\show\csname glo@glSDetoklabel{#1}@Symbol\endcsname  
}
```

`\showgloSymbolPlural`

```
\showgloSymbolPlural{<label>}
```

```
\newcommand*{\showgloSymbolPlural}[1]{%  
  \expandafter\show\csname glo@glSDetoklabel{#1}@SymbolPlural\endcsname  
}
```

`\showgloShort`

```
\showgloShort{<label>}
```

```
\newcommand*{\showgloShort}[1]{%  
  \expandafter\show\csname glo@glSDetoklabel{#1}@Short\endcsname  
}
```

`\showgloLong`

```
\showgloLong{<label>}
```

```
\newcommand*{\showgloLong}[1]{%  
  \expandafter\show\csname glo@glSDetoklabel{#1}@Long\endcsname  
}
```

`\showgloindex`

```
\showgloindex{<label>}
```

```
\newcommand*{\showgloindex}[1]{%  
  \expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname  
}
```

`\showgloflag`

```
\showgloflag{<label>}
```

```
\newcommand*{\showgloflag}[1]{%  
  \expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname  
}
```

`\showgloloclist`

```
\showgloloclist{<label>}
```

```
\newcommand*{\showgloloclist}[1]{%  
  \expandafter\show\csname glo@\glsdetoklabel{#1}@loclist\endcsname  
}
```

`\showglofield`

```
\showglofield{<label>}{<field>}
```

```
\newcommand*{\showglofield}[2]{%  
  \csshow{glo@\glsdetoklabel{#1}@#2}%  
}
```

`\showacronymlists`

```
\showacronymlists
```

Show list of glossaries that have been flagged as a list of acronyms.

```
\newcommand*{\showacronymlists}{%  
  \show\@glsacronymlists  
}
```

`\showglossaries`

```
\showglossaries
```

Show list of defined glossaries.

```
\newcommand*{\showglossaries}{%  
  \show\@glo@types  
}
```

`\showglossaryin`

```
\showglossaryin{<glossary-label>}
```

Show the ‘in’ extension for the given glossary.

```
\newcommand*{\showglossaryin}[1]{%
  \expandafter\show\csname @glotype@#1in\endcsname
}
```

`\showglossaryout`

```
\showglossaryout{<glossary-label>}
```

Show the ‘out’ extension for the given glossary.

```
\newcommand*{\showglossaryout}[1]{%
  \expandafter\show\csname @glotype@#1out\endcsname
}
```

`\showglossarytitle`

```
\showglossarytitle{<glossary-label>}
```

Show the title for the given glossary.

```
\newcommand*{\showglossarytitle}[1]{%
  \expandafter\show\csname @glotype@#1title\endcsname
}
```

`\showglossarycounter`

```
\showglossarycounter{<glossary-label>}
```

Show the counter for the given glossary.

```
\newcommand*{\showglossarycounter}[1]{%
  \expandafter\show\csname @glotype@#1counter\endcsname
}
```

`\showglossaryentries`

```
\showglossaryentries{<glossary-label>}
```

Show the list of entry labels for the given glossary.

```
\newcommand*{\showglossaryentries}[1]{%
  \expandafter\show\csname glolist@#1\endcsname
}
```

2.21 Compatibility with version 2.07 and below

The backward-compatibility options have been removed in version 4.50. Use the rollback feature instead if backward-compatibility is required. Bear in mind that v2.07 was released 2010-07-10 and v3.07 was released 2013-07-05. Any old documents from that long ago will probably have problems with recent kernels.

In order to fix some bugs in v3.0, it was necessary to change the way information is written to the `glo` file, which also meant a change in the format of the Xindy style file. The compatibility option is meant for documents that use a customised Xindy style file with `\noist`. With the compatibility option, hopefully `xindy` will still be able to process the old document, but the bugs will remain. The issues in versions 2.07 and below:

- With `xindy`, the counter used by the entry was hard-coded into the Xindy style file. This meant that you couldn't use the `counter` to swap counters.
- With both `xindy` and `makeindex`, if used with `hyperref` and `\theH{counter}` was different to `\thecounter`, the link in the location number would be undefined.

3 Glossary Styles

3.1 Glossary hyper-navigation definitions (`glossary-hypernav.sty`)

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-hypernav-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-hypernav-2021-11-01.sty}
\DeclareRelease{v4.52}{2022-11-03}{glossary-hypernav-2022-11-03.sty}
\DeclareCurrentRelease{v4.56}{2025-03-03}
```

Package Definition:

```
\ProvidesPackage{glossary-hypernav}[2025/03/03 v4.56 (NLCT)]
```

The commands defined in this package are provided to help navigate around the groups within a glossary (see [subsection 2.16.](#)) `\printglossary` (and `\printglossaries`) set `\@glo@type` to the label of the current glossary. This is used to create a unique `hypertarget` in the event of multiple glossaries.

```
\glsnavhyperlink[<type>]{<label>}{<text>}
```

This command makes `<text>` a hyperlink to the glossary group whose label is given by `<label>` for the glossary given by `<type>`.

`\glsnavhyperlink` The optional argument is the glossary type and can be used to make the group label unique in case the same group occurs in multiple glossaries. The optional argument was `\@glo@type` but is now `\currentglossary` because that's less likely to change within the glossary. It's unlikely that the optional argument will be used to set it to anything other than `\currentglossary` but it makes it possible to hyperlink to the group from outside of the glossary.

```
\newcommand*\glsnavhyperlink[3][\currentglossary]{%
\protected@edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%
\@glslink{\glsnavhyperlinkname{#1}{#2}}{#3}%
}
```


`\glsnavhyperlinkname` Expands to the hypertarget name. The first argument is the glossary type. The second argument is the group label.

```
\newcommand*{\glsnavhyperlinkname}[2]{glsn:#1@#2}
```

Version 4.53: reimplemented list of hypertarget groups as a sequence. Previously, the targets were stored as comma-separated lists in `\@gls@hypergrouplist@<glossary>` (where `<glossary>` is the glossary label). They are now stored in a sequence `\g_glossaries_<glossary>_hypergroups_seq`.

```
\ExplSyntaxOn
```

```
\glossaries_add_hypergroup:nn{<glossary>}{<target>}
```

Add a target to the sequence unless it's already contained in it:

```
\cs_new:Npn \glossaries_add_hypergroup:nn #1 #2
{
  \seq_if_exist:cTF
  { g_glossaries_ #1 _hypergroups_seq }
  {
    \seq_if_in:cnF
    { g_glossaries_ #1 _hypergroups_seq } { #2 }
    {
      \seq_gput_right:cn
      { g_glossaries_ #1 _hypergroups_seq }
      { #2 }
    }
  }
}
\cs_generate_variant:Nn \glossaries_add_hypergroup:nn { nx }
```

Check if the given target has been added to the sequence to avoid duplicates.

```
\cs_new:Npn \glossaries_if_hypergroup:nnTF #1 #2 #3 #4
{
  \seq_if_exist:cTF
  { g_glossaries_ #1 _hypergroups_seq }
  {
    \seq_if_in:cnTF
    { g_glossaries_ #1 _hypergroups_seq } { #2 }
    { #3 } { #4 }
  }
  { #4 }
}
\cs_generate_variant:Nn \glossaries_if_hypergroup:nnTF { nxTF }
```

Only true part:

```
\cs_new:Npn \glossaries_if_hypergroup:nnT #1 #2 #3
{
  \glossaries_if_hypergroup:nnTF { #1 } { #2 } { #3 } { }
}
\cs_generate_variant:Nn \glossaries_if_hypergroup:nnT { nxT }
```

Only false part:

```
\cs_new:Npn \glossaries_if_hypergroup:nnF #1 #2 #3
{
  \glossaries_if_hypergroup:nnTF { #1 } { #2 } { } { #3 }
}
\cs_generate_variant:Nn \glossaries_if_hypergroup:nnF { nxF }
```

```
\glsnavhypertarget[type]{label}{text}
```

`\glsnavhypertarget`

This command makes *text* a hypertarget for the glossary group whose label is given by *label* in the glossary given by *type*. If *type* is omitted, `\gls@type` is used which is set by `\printglossary` to the current glossary label. The optional argument complicates things for `glossaries-extra`.

```
\newcommand*\glsnavhypertarget{\protect\@gls@navhypertarget}
```

`\@gls@navhypertarget`

```
\newcommand*\@gls@navhypertarget[3][\currentglossary]{%
  \glsnavhypertarget{#1}{#2}{#3}%
}
```

`\glsnavhypergroupdotarget` Provided to make it easier for `glossaries-extra` to adjust the target.

```
\newcommand{\glsnavhypergroupdotarget}[3]{
  \glsstarget{\glsnavhyperlinkname{#1}{#2}}{#3}%
}
```

The actual code is now in an internal command that doesn't have an optional argument, which makes it easier to save and restore the original behaviour.

`\@glsnavhypertarget`

```
\newcommand*\@glsnavhypertarget[3]{%
```

Add this group to the aux file for re-run check.

```
\protected@write\@auxout{\string\@gls@hypergroup{#1}{#2}}%
```

Add the target.

```
\glsnavhypergroupdotarget { #1 } { #2 } { #3 }
```

Check list of known groups to determine if a re-run is required.

```
\glossaries_if_hypergroup:nxF { #1 } { #2 }
{
```

This group was not included in the list, so issue a warning.

```

\GlossariesWarningNoLine{Navigation ~ panel ~
  for ~ glossary ~ type ~ '#1'^^Jmissing ~ group ~ '#2'}%
\gdef\gls@hypergrouprerun{%
  \GlossariesWarningNoLine{Navigation ~ panel ~
    has ~ changed. ~ Rerun ~ LaTeX}}%
}
}

```

`\gls@hypergrouprerun` Give a warning at the end if re-run required

```

\let\gls@hypergrouprerun\relax
\AtEndDocument{\gls@hypergrouprerun}

```

`\@gls@hypergroup` This adds to (or creates) the command `\@gls@hypergrouplist@{glossary type}` which lists all groups for a given glossary, so that the navigation bar only contains those groups that are present. However it requires at least 2 runs to ensure the information is up-to-date.

```

\newcommand*{\@gls@hypergroup}[2]{
  \glossaries_add_hypergroup:nx { #1 } { #2 }
}

```

The `\glsnavigation` command displays a simple glossary group navigation. The symbol and number elements are defined separately, so that they can be suppressed if need be. (In earlier versions this command will produce a link to all 28 groups, but some groups may not be defined if there are groups that do not contain any terms, in which case you will get an undefined hyperlink warning. Version 1.14 changed this to only use labels for groups that are present.) Now for the whole navigation bit:

`\glsnavigation`

```

\newcommand*{\glsnavigation}{%
  \def\@gls@between{}%
  \seq_map_inline:cn { g_glossaries_ \currentglossary _hypergroups_seq }
  {
    \@gls@between
    \glsnavigationitem { ##1 }
    \let\@gls@between\glshypernavsep
  }%
}

```

`\glsnavigationitem`

```

\newcommand{\glsnavigationitem}[1]{%
  \@gls@getgrouptitle { #1 } { \@gls@grptitle }
  \exp_args:Nno \glsnavhyperlink { #1 } { \@gls@grptitle }
}

```

`\glshypernavsep` Separator for the hyper navigation bar.

```

\newcommand*{\glshypernavsep}{\space\textbar\space}

```

The `\glssymbolnav` produces a simple navigation set of links for just the symbol and number groups. This used to be used at the start of `\glsnavigation`. This command is no longer needed.

```
\glssymbolnav
\newcommand*\glssymbolnav{%
  \glsnavhyperlink{glssymbols}{\glsgetgrouptitle{glssymbols}}%
  \glshypernavsep
  \glsnavhyperlink{glsnumbers}{\glsgetgrouptitle{glsnumbers}}%
  \glshypernavsep
}
\ExplSyntaxOff
```

3.2 In-line Style (`glossary-inline.sty`)

This defines an in-line style where the entries are comma-separated with just the name and description displayed.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-inline-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-inline-2021-11-01.sty}
\DeclareCurrentRelease{v4.56}{2025-03-03}
```

Declare package:

```
\ProvidesPackage{glossary-inline}[2025/03/03 v4.56 (NLCT)]
```

`inline` Define the inline style.

```
\newglossarystyle{inline}{%
```

Start of glossary sets up first empty separator between entries. (This is then changed by `\glossentry`)

```
\renewenvironment{theglossary}%
  {%
    \def\gls@inlinesep{}%
    \def\gls@inlinesubsep{}%
    \def\gls@inlinepostchild{}%
  }%
  {\glspostinline}%
```

No header:

```
\renewcommand*\glossaryheader{}%
```

No group headings (if heading is required, add `\glsinlinedopostchild` to start definition in case heading follows a child entry):

```
\renewcommand*\glsgroupheading}[1]{}%
```

Just display separator followed by name and description:

```
\renewcommand{\glossentry}[2]{%
  \glsinlinedopostchild
```

```

\gls@inlinesep
\glsentryitem{##1}%
\glsinlinenameformat{##1}{%
  \glossentryname{##1}%
}%
\ifglsdescsuppressed{##1}%
{%
  \glsinlineemptydescformat
  {%
    \glossentrysymbol{##1}%
  }%
  {%
    ##2%
  }%
}%
{%
  \ifglshasdesc{##1}%
  {\glsinlinedescformat{\glossentrydesc{##1}}{\glossentrysymbol{##1}}{##2}}%
  {\glsinlineemptydescformat{\glossentrysymbol{##1}}{##2}}%
}%
\glsinlineifhaschildren{##1}%
{%
  \glsresetsubentrycounter
  \glsinlineparentchildseparator
  \def\gls@inlinesubsep{}%
  \def\gls@inlinepostchild{\glsinlinepostchild}%
}%
{%
  {}%
  \def\gls@inlinesep{\glsinlineseparator}%
}%

```

Sub-entries display description:

```

\renewcommand{\subglossentry}[3]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{%
    \glossentryname{##2}}%
  \glssubentryitem{##2}%
  \glsinlinesubdescformat{\glossentrydesc{##2}}{\glossentrysymbol{##2}}{##3}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%

```

Nothing special between groups:

```

\renewcommand*{\glsgroupskip}{}%
}

```

\glsinlineifhaschildren

```

\newcommand{\glsinlineifhaschildren}[3]{%
  \ifglshaschildren{#1}{#2}{#3}%
}

```

\glsinlinedopostchild

```

\newcommand*\glsinlinedopostchild{%
  \gls@inlinepostchild
  \def\gls@inlinepostchild{}%
}

\glsinlineseparator Separator to use between entries.
\newcommand*\glsinlineseparator}{;\space}

\glsinlinesubseparator Separator to use between sub-entries.
\newcommand*\glsinlinesubseparator}{,\space}

\glsinlineparentchildseparator Separator to use between parent and children.
\newcommand*\glsinlineparentchildseparator}{:\space}

\glsinlinepostchild Hook to use between child and next entry
\newcommand*\glsinlinepostchild}{

\glspostinline Terminator for inline glossary.
\newcommand*\glspostinline}{\glspostdescription\space}

\glsinlinenameformat Formats the name of the entry (first argument label, second argument name):
\newcommand*\glsinlinenameformat}[2]{\glstarget{#1}{#2}}

\glsinlinedescformat Formats the entry's description, symbol and location list:
\newcommand*\glsinlinedescformat}[3]{\space#1}

\glsinlineemptydescformat Formats the entry's symbol and location list when the description is empty:
\newcommand*\glsinlineemptydescformat}[2]{

\glsinlinesubnameformat Formats the name of the subentry (first argument label, second argument name):
\newcommand*\glsinlinesubnameformat}[2]{\glstarget{#1}{}}

\glsinlinesubdescformat Formats the subentry's description, symbol and location list:
\newcommand*\glsinlinesubdescformat}[3]{#1}

```

3.3 List Style (glossary-list.sty)

The `glossary-list` style file defines glossary styles that use the `description` environment. Note that since the entry name is placed in the optional argument to the `\item` command, it will appear in a bold font by default.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-list-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-list-2021-11-01.sty}
\DeclareCurrentRelease{v4.56}{2025-03-03}
```

Declare package:

```
\ProvidesPackage{glossary-list}[2025/03/03 v4.56 (NLCT)]
```

`\indexspace` There are a few classes that don't define `\indexspace`, so provide a definition if it hasn't been defined.

```
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
```

`\glslistgroupheaderfmt` Provide a way of adjusting the format of the group headings.

```
\newcommand*{\glslistgroupheaderfmt}[1]{#1}
```

`\glslistnavigationitem` Provide a way of adjusting the format of the navigation header. This puts the navigation line inside the optional argument of `item` to prevent unwanted space occurring at the start, but this can cause a problem if the navigation line is too long. With this command, it makes it easier for the user to customise the style without having to remember to modify `\glossaryheader` after the style has been set.

```
\newcommand*{\glslistnavigationitem}[1]{\item[#1]}
```

`\glslistinit` Provide a way to ensure appropriate expansion of the name if `entrycounter` is used with `getttitlestring`:

```
\newcommand{\glslistinit}{%
  \ifdef\GetTitleStringDisableCommands
  {%
    \GetTitleStringSetup{expand}%
    \GetTitleStringDisableCommands{%
      \let\glsentryitem\@gobble
      \let\glstarget\@secondoftwo
      \let\glossentryname\glslistexpandedname
      \let\glslistgroupheaderfmt\@firstofone
      \let\glsgetgrouptitle\@firstofone
      \let\glsnavhypertarget\@secondoftwo
      \let\glsnavigation\relax
    }%
  }%
  {}%
}
```

`\glslistexpandedname`

```
\newcommand{\glslistexpandedname}[1]{%
  \glsunexpandedfieldvalue{#1}{name}%
}
```

`list` The list glossary style uses the `description` environment. The group separator `\glsgroupskip` is redefined as `\indexspace` which produces a gap between groups. The glossary heading and the group headings do nothing. Sub-entries immediately follow the main entry without the sub-entry name. This style does not use the entry's symbol. This is used as the default style for the `glossaries` package.

```
\newglossarystyle{list}{%
```

Use description environment:

```
\renewenvironment{theglossary}%  
  {\glslistinit\begin{description}}{\end{description}}%
```

No header at the start of the environment:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries start a new item in the list:

```
\renewcommand*{\glossentry}[2]{%  
  \item[\glsentryitem{##1}%  
    \glstarget{##1}{\glossentryname{##1}}]  
    \glossentrydesc{##1}\glspostdescription\space ##2}%
```

Sub-entries continue on the same line:

```
\renewcommand*{\subglossentry}[3]{%  
  \glssubentryitem{##2}%  
  \glstarget{##2}{\strut}\space  
  \glossentrydesc{##2}\glspostdescription\space ##3.}%
```

Add vertical space between groups:

```
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%  
}
```

listgroup The listgroup style is like the list style, but the glossary groups have headings.

```
\newglossarystyle{listgroup}{%
```

Base it on the list style:

```
\setglossarystyle{list}%
```

Each group has a heading:

```
\renewcommand*{\glsgroupheading}[1]{%  
  \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
```

listhypergroup The listhypergroup style is like the listgroup style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{listhypergroup}{%
```

Base it on the list style:

```
\setglossarystyle{list}%
```

Add navigation links at the start of the environment.

```
\renewcommand*{\glossaryheader}{%  
  \glslistnavigationitem{\glsnavigation}}%
```

Each group has a heading with a hypertarget:

```
\renewcommand*{\glsgroupheading}[1]{%  
  \item[\glslistgroupheaderfmt  
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}]}
```


altlist The altlist glossary style is like the list style, but places the description on a new line. Sub-entries follow in separate paragraphs without the sub-entry name. This style does not use the entry's symbol.

```
\newglossarystyle{altlist}{%
```

Base it on the list style:

```
\setglossarystyle{list}%
```

Main (level 0) entries start a new item in the list with a line break after the entry name:

```
\renewcommand*{\glossentry}[2]{%
\item[\glsentryitem{##1}%
\glstarget{##1}{\glossentryname{##1}}]%
```

Version 3.04 changed `\newline` to the following paragraph break stuff (thanks to Daniel Gebhardt for supplying the fix) to prevent a page break occurring at this point.

```
\mbox{}\par\nobreak\@afterheading
\glossentrydesc{##1}\glspostdescription\space ##2}%
```

Sub-entries start a new paragraph:

```
\renewcommand{\subglossentry}[3]{%
\par
\glsentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space ##3}%
}
```

altlistgroup The altlistgroup glossary style is like the altlist style, but the glossary groups have headings.

```
\newglossarystyle{altlistgroup}{%
```

Base it on the altlist style:

```
\setglossarystyle{altlist}%
```

Each group has a heading:

```
\renewcommand*{\glsgroupheading}[1]{%
\item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
```

altlisthypergroup The altlisthypergroup glossary style is like the altlistgroup style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{altlisthypergroup}{%
```

Base it on the altlist style:

```
\setglossarystyle{altlist}%
```

Add navigation links at the start of the environment.

```
\renewcommand*{\glossaryheader}{%
\glslistnavigationitem{\glsnavigation}}%
```

Each group has a heading with a hypertext:

```
\renewcommand*{\glsgroupheading}[1]{%
\item[\glslistgroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}]}
```

`listdotted` The `listdotted` glossary style was supplied by Axel Menzel. I've modified it slightly so that the distance from the start of the name to the end of the dotted line is specified by `\glslistdottedwidth`. Note that this style ignores the page numbers as well as the symbol. Sub-entries are displayed in the same way as top-level entries.

```
\newglossarystyle{listdotted}{%
```

Base it on the list style:

```
\setglossarystyle{list}{%
```

Each main (level 0) entry starts a new item:

```
\renewcommand*{\glossentry}[2]{%
  \item[\makebox[\glslistdottedwidth][l]{%
    \glsentryitem{##1}%
    \glstarget{##1}{\glossentryname{##1}}%
    \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##1}}%
```

Sub entries have the same format as main entries:

```
\renewcommand*{\subglossentry}[3]{%
  \item[\makebox[\glslistdottedwidth][l]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\glossentryname{##2}}%
    \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##2}}%
}
```

```
\glslistdottedwidth
```

```
\newlength\glslistdottedwidth
\setlength{\glslistdottedwidth}{.5\hsize}
```

`sublistdotted` This style is similar to the `glostylelistdotted` style, except that the main entries just have the name displayed.

```
\newglossarystyle{sublistdotted}{%
```

Base it on the `listdotted` style:

```
\setglossarystyle{listdotted}{%
```

Main (level 0) entries just display the name:

```
\renewcommand*{\glossentry}[2]{%
  \item[\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}}]}%
```

3.4 Glossary Styles using `longtable` (`glossary-long.sty`)

The glossary styles defined in the `glossary-long` package used the `longtable` environment in the glossary.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-long-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-long-2021-11-01.sty}
\DeclareCurrentRelease{v4.56}{2025-03-03}
```

Declare package:

```
\ProvidesPackage{glossary-long}[2025/03/03 v4.56 (NLCT)]
```

Requires the longtable package:

```
\RequirePackage{longtable}
```

`\glsdescwidth` This is a length that governs the width of the description column. (There's a chance that the user may specify `nolong` and then load `glossary-long` later, in which case `\glsdescwidth` may have already been defined by `glossary-super`. The same goes for `\glspagelistwidth`.)

```
\@ifundefined{glsdescwidth}{%  
  \newlength{glsdescwidth}  
  \setlength{glsdescwidth}{0.6\hsize}  
}{}
```

`\glspagelistwidth` This is a length that governs the width of the page list column.

```
\@ifundefined{glspagelistwidth}{%  
  \newlength{glspagelistwidth}  
  \setlength{glspagelistwidth}{0.1\hsize}  
}{}
```

`long` The long glossary style command which uses the `longtable` environment:

```
\newglossarystyle{long}{%
```

Use `longtable` with two columns:

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{lp{glsdescwidth}}}%  
  {\end{longtable}}%
```

Do nothing at the start of the environment:

```
\renewcommand*{\glossaryheader}{}%
```

No heading between groups:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries displayed in a row:

```
\renewcommand{\glossentry}[2]{%  
  \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &  
  \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline  
}%
```

Sub entries displayed on the following row without the name:

```
\renewcommand{\subglossentry}[3]{%  
  &  
  \glssubentryitem{##2}%  
  \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space  
  ##3\tabularnewline  
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip{ & \tabularnewline}%
\fi
}
```

`longborder` The `longborder` style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{longborder}{%
```

Base it on the `glostylelong` style:

```
\setglossarystyle{long}%
```

Use `longtable` with two columns with vertical lines between each column:

```
\renewenvironment{theglossary}{%
  \begin{longtable}{|l|p{\glsdescwidth}|}{\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
\renewcommand*\glossaryheader{\hline\endhead\hline\endfoot}%
}
```

`longheader` The `longheader` style is like the `long` style but with a header:

```
\newglossarystyle{longheader}{%
```

Base it on the `glostylelong` style:

```
\setglossarystyle{long}%
```

Set the table's header:

```
\renewcommand*\glossaryheader{%
  \bfseries \entryname & \bfseries \descriptionname\tabularnewline\endhead}%
}
```

`longheaderborder` The `longheaderborder` style is like the `long` style but with a header and border:

```
\newglossarystyle{longheaderborder}{%
```

Base it on the `glostylelongborder` style:

```
\setglossarystyle{longborder}%
```

Set the table's header and add horizontal line to table's foot:

```
\renewcommand*\glossaryheader{%
  \hline\bfseries \entryname & \bfseries
  \descriptionname\tabularnewline\hline
\endhead
\hline\endfoot}%
}
```

`long3col` The `long3col` style is like `long` but with 3 columns

```
\newglossarystyle{long3col}{%
```

Use a `longtable` with 3 columns:

```
\renewenvironment{theglossary}%
  {\begin{longtable}[lp{\glsdescwidth}p{\glspagelistwidth}}%
  {\end{longtable}}%
```

No table header:

```
\renewcommand*{\glossaryheader}{}%
```

No headings between groups:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} & ##2\tabularnewline
}%
```

Sub-entries on a separate row (no name, description in second column, page list in third column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
\fi
}
```

`long3colborder` The `long3colborder` style is like the `long3col` style but with a border:

```
\newglossarystyle{long3colborder}{}%
```

Base it on the `glostylelong3col` style:

```
\setglossarystyle{long3col}{}%
```

Use a `longtable` with 3 columns with vertical lines around them:

```
\renewenvironment{theglossary}%
  {\begin{longtable}[|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
  {\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
```

`long3colheader` The `long3colheader` style is like `long3col` but with a header row:

```
\newglossarystyle{long3colheader}{%
Base it on the glostylelong3col style:
\setglossarystyle{long3col}%
Set the table's header:
\renewcommand*{\glossaryheader}{%
\bfseries\entryname&\bfseries\descriptionname&
\bfseries\pagelistname\tabularnewline\endhead}%
}
```

`long3colheaderborder` The `long3colheaderborder` style is like the above but with a border

```
\newglossarystyle{long3colheaderborder}{%
Base it on the glostylelong3colborder style:
\setglossarystyle{long3colborder}%
Set the table's header and add horizontal line at table's foot:
\renewcommand*{\glossaryheader}{%
\hline
\bfseries\entryname&\bfseries\descriptionname&
\bfseries\pagelistname\tabularnewline\hline\endhead
\hline\endfoot}%
}
```

`long4col` The `long4col` style has four columns where the third column contains the value of the associated symbol key.

```
\newglossarystyle{long4col}{%
Use a longtable with 4 columns:
\renewenvironment{theglossary}%
{\begin{longtable}{llll}}%
{\end{longtable}}%
No table header:
\renewcommand*{\glossaryheader}{}%
No group headings:
\renewcommand*{\glsgroupheading}[1]{}%
Main (level 0) entries on a single row (name in first column, description in
second column, symbol in third column, page list in last column):
\renewcommand{\glossentry}[2]{%
\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1} &
\glossentrysymbol{##1} &
##2\tabularnewline
}%
```

Sub entries on a single row with no name (description in second column, symbol in third column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  \glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for nogroupskip must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip{ & & \tabularnewline}%
\fi
}
```

`long4colheader` The `long4colheader` style is like `long4col` but with a header row.

```
\newglossarystyle{long4colheader}{%
Base it on the glostylelong4col style:
  \setglossarystyle{long4col}%
Table has a header:
  \renewcommand*\glossaryheader{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
```

`long4colborder` The `long4colborder` style is like `long4col` but with a border.

```
\newglossarystyle{long4colborder}{%
Base it on the glostylelong4col style:
  \setglossarystyle{long4col}%
Use a longtable with 4 columns surrounded by vertical lines:
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|}}%
    {\end{longtable}}%
Add horizontal lines to the head and foot of the table:
  \renewcommand*\glossaryheader{\hline\endhead\hline\endfoot}%
}
```

`long4colheaderborder` The `long4colheaderborder` style is like the above but with a border.

```
\newglossarystyle{long4colheaderborder}{%
Base it on the glostylelong4col style:
  \setglossarystyle{long4col}%
```

Use a `longtable` with 4 columns surrounded by vertical lines:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{|l|l|l|l|}}%
  {\end{longtable}}%
```

Add table header and horizontal line at the table's foot:

```
\renewcommand*{\glossaryheader}{%
  \hline\bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\hline\endhead
  \hline\endfoot}%
}
```

`altlong4col` The `altlong4col` style is like the `long4col` style but can have multiline descriptions and page lists.

```
\newglossarystyle{altlong4col}{%
```

Base it on the `glostylelong4col` style:

```
\setglossarystyle{long4col}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}%
  {\end{longtable}}%
}
```

`altlong4colheader` The `altlong4colheader` style is like `altlong4col` but with a header row.

```
\newglossarystyle{altlong4colheader}{%
```

Base it on the `glostylelong4colheader` style:

```
\setglossarystyle{long4colheader}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}%
  {\end{longtable}}%
}
```

`altlong4colborder` The `altlong4colborder` style is like `altlong4col` but with a border.

```
\newglossarystyle{altlong4colborder}{%
```

Base it on the `glostylelong4colborder` style:

```
\setglossarystyle{long4colborder}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{|l|lp{\glsdescwidth}|lp{\glspagelistwidth}|}%
  {\end{longtable}}%
}
```


`altlong4colheaderborder` The `altlong4colheaderborder` style is like the above but with a header as well as a border.

```
\newglossarystyle{altlong4colheaderborder}{%
```

Base it on the `glostylolong4colheaderborder` style:

```
\setglossarystyle{long4colheaderborder}{%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}{%
  {\begin{longtable}{|l|p{\glsdescwidth}|l|p{\glspagelistwidth}|}%
  {\end{longtable}}}%
}
```

3.5 Glossary Styles using `longtable` and `booktabs` (`glossary-longbooktabs.sty`)

The styles here are based on David Carlisle's patch at <http://tex.stackexchange.com/a/56890>

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-longbooktabs-2020-03-19.sty}
```

```
\DeclareRelease{v4.49}{2021-11-01}{glossary-longbooktabs-2021-11-01.sty}
```

```
\DeclareCurrentRelease{v4.56}{2025-03-03}
```

Declare package:

```
\ProvidesPackage{glossary-longbooktabs}[2025/03/03 v4.56 (NLCT)]
```

Requires `booktabs` package:

```
\RequirePackage{booktabs}
```

and the base packages for long styles:

```
\RequirePackage{glossary-long}
```

```
\RequirePackage{glossary-longragged}
```

(`longtable` and `array` loaded by those packages).

`long-booktabs` The `long-booktabs` style is similar to the `longheader` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{long-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

As with the `longheader` style, use the `long` style as a base.

```
\setglossarystyle{long}{%
```

Add a header with rules.

```
\renewcommand*\glossaryheader}{%
  \toprule \bfseries \entryname & \bfseries
  \descriptionname\tabularnewline\midrule\endhead
  \bottomrule\endfoot}%
```

Check for the `nogroupskip` package option. If there should be a gap between groups, insert the penalty and the vertical space. The check for `nogroupskip` should occur outside `\glsgroupskip` to be on the safe side.

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip}{}%
\else
  \renewcommand*\glsgroupskip}{\glspenaltygroupskip}%
\fi
}
```

`long3col-booktabs` The `long3col-booktabs` style is similar to the `long3colheader` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{long3col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the `long3col` style as a base.

```
\setglossarystyle{long3col}%
```

Add a header with rules.

```
\renewcommand*\glossaryheader}{%
  \toprule \bfseries \entryname &
  \bfseries \descriptionname &
  \bfseries \pagelistname
  \tabularnewline\midrule\endhead
  \bottomrule\endfoot}%
```

Check for the `nogroupskip` package option. If there should be a gap between groups, insert the penalty and the vertical space. The check for `nogroupskip` should occur outside `\glsgroupskip` to be on the safe side.

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip}{}%
\else
  \renewcommand*\glsgroupskip}{\glspenaltygroupskip}%
\fi
}
```

`long4col-booktabs` The `long4col-booktabs` style is similar to the `long4colheader` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{long4col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the `long4col` style as a base.

```
\setglossarystyle{long4col}%
```

Add a header with rules.

```
\renewcommand*{\glossaryheader}{%
  \toprule \bfseries \entryname &
  \bfseries \descriptionname &
  \bfseries \symbolname &
  \bfseries \pagelistname
  \tabularnewline\midrule\endhead
  \bottomrule\endfoot}%
```

Check for the `nogroupskip` package option. If there should be a gap between groups, insert the penalty and the vertical space. The check for `nogroupskip` should occur outside `\glsgroupskip` to be on the safe side.

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}
```

`altlong4col-booktabs` The `altlong4col-booktabs` style is similar to the `altlong4colheader` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{altlong4col-booktabs}{%
```

The patch `\glspatchLToutput` is already applied in `long4col-booktabs` and so doesn't need to be here.

```
\glspatchLToutput
```

Use the `long4col-booktabs` style as a base.

```
\setglossarystyle{long4col-booktabs}%
```

Change the column specifications:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{lp{\glstdescwidth}lp{\glspagelistwidth}}}%
  {\end{longtable}}%
}
```

Ragged styles.

`longragged-booktabs` The `longragged-booktabs` style is similar to the `longragged` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{longragged-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the long-booktabs style as a base.

```
\setglossarystyle{long-booktabs}%
```

Adjust the column specification.

```
\renewenvironment{theglossary}%
  {\begin{longtable}{1>{\raggedright}p{\glsdescwidth}}}%
  {\end{longtable}}%
}
```

longragged3col-booktabs The longragged3col-booktabs style is similar to the longragged3col style but uses the booktabs rules and patches longtable to check for group skip occurring at a page break.

```
\newglossarystyle{longragged3col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the long3col-booktabs style as a base.

```
\setglossarystyle{long3col-booktabs}%
```

Adjust the column specification.

```
\renewenvironment{theglossary}%
  {\begin{longtable}{1>{\raggedright}p{\glsdescwidth}%
    >{\raggedright}p{\glspagelistwidth}}}%
  {\end{longtable}}%
}
```

altlongragged4col-booktabs The altlongragged4col-booktabs style is similar to the altlongragged4col style but uses the booktabs rules and patches longtable to check for group skip occurring at a page break.

```
\newglossarystyle{altlongragged4col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the altlong4col-booktabs style as a base.

```
\setglossarystyle{altlong4col-booktabs}%
```

Adjust the column specification.

```
\renewenvironment{theglossary}%
  {\begin{longtable}{1>{\raggedright}p{\glsdescwidth}1%
    >{\raggedright}p{\glspagelistwidth}}}%
  {\end{longtable}}%
}
```

```

\glsLTpenaltycheck
  \newcommand*{\glsLTpenaltycheck}{%
    \ifnum\outputpenalty=-50\vskip-\normalbaselineskip\relax\fi
  }

```

```

\glspenaltygroupskip
  \newcommand{\glspenaltygroupskip}{%
    \noalign{\penalty-50\vskip\normalbaselineskip}}

```

```

\glsrestoreLToutput Provide a way of restoring \LT@output for the user.
  \let\@gls@org@LT@output\LT@output
  \newcommand*{\glsrestoreLToutput}{\let\LT@output\@gls@org@LT@output}

```

This is David's patch, but I've replaced the hard-coded values with `\glsLTpenaltycheck` to make it easier to adjust.

```

\glspatchLToutput
  \newcommand*{\glspatchLToutput}{%
    \renewcommand*{\LT@output}{%
      \ifnum\outputpenalty <-\@Mi
        \ifnum\outputpenalty > -\LT@end@pen
          \LT@err{floats and marginpars not allowed in a longtable}\@ehc
        \else
          \setbox\z@\vbox{\unvbox\@cclv}%
          \ifdim \ht\LT@lastfoot>\ht\LT@foot
            \dimen@pagegoal
            \advance\dimen@-\ht\LT@lastfoot
            \ifdim\dimen@<\ht\z@
              \setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%
              \@makecol
              \@outputpage
              \setbox\z@\vbox{\box\LT@head\glsLTpenaltycheck}%
            \fi
          \fi
          \global\@colroom\@colht
          \global\vsizel\@colht
          {\unvbox\z@\box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%
        \fi
      \else
        \setbox\@cclv\vbox{\unvbox\@cclv\copy\LT@foot\vss}%
        \@makecol
        \@outputpage
        \global\vsizel\@colroom
        \copy\LT@head
        \glsLTpenaltycheck
        \nobreak
      \fi
    }%
  }

```

3.6 Glossary Styles using longtable (glossary-longragged.sty)

The glossary styles defined in the `glossary-longragged` package used the `longtable` environment in the `glossary` and use ragged right formatting for the multiline columns.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-longragged-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-longragged-2021-11-01.sty}
\DeclareCurrentRelease{v4.56}{2025-03-03}
```

Declare package:

```
\ProvidesPackage{glossary-longragged}[2025/03/03 v4.56 (NLCT)]
```

Requires the `array` package:

```
\RequirePackage{array}
```

Requires the `longtable` package:

```
\RequirePackage{longtable}
```

`\glsdescwidth` This is a length that governs the width of the description column. This may have already been defined.

```
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{}
```

`\glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined.

```
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{}
```

`longragged` The `longragged` glossary style is like the `long` but uses ragged right formatting for the description column.

```
\newglossarystyle{longragged}{%
```

Use `longtable` with two columns:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{1>{\raggedright}p{glsdescwidth}}}%
  {\end{longtable}}%
```

Do nothing at the start of the environment:

```
\renewcommand*{\glossaryheader}{}%
```

No heading between groups:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries displayed in a row:

```
\renewcommand{\glossentry}[2]{%
  \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1}\glspostdescription\space ##2%
  \tabularnewline
}%
```

Sub entries displayed on the following row without the name:

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}%
  \glspostdescription\space ##3%
  \tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
\fi
}
```

`longraggedborder` The `longraggedborder` style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{longraggedborder}{%
```

Base it on the `glostylelongragged` style:

```
\setglossarystyle{longragged}%
```

Use `longtable` with two columns with vertical lines between each column:

```
\renewenvironment{theglossary}{%
  \begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|}%
  {\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
```

`longraggedheader` The `longraggedheader` style is like the `longragged` style but with a header:

```
\newglossarystyle{longraggedheader}{%
```

Base it on the `glostylelongragged` style:

```
\setglossarystyle{longragged}%
```

Set the table's header:

```
\renewcommand*{\glossaryheader}{%
  \bfseries \entryname & \bfseries \descriptionname
  \tabularnewline\endhead}%
}
```

`longraggedheaderborder` The `longraggedheaderborder` style is like the `longragged` style but with a header and border:

```
\newglossarystyle{longraggedheaderborder}{%
```

Base it on the `glostyplelongraggedborder` style:

```
\setglossarystyle{longraggedborder}{%
```

Set the table's header and add horizontal line to table's foot:

```
\renewcommand*{\glossaryheader}{%
  \hline\bfseries \entryname & \bfseries \descriptionname
  \tabularnewline\hline
  \endhead
  \hline\endfoot}%
}
```

`longragged3col` The `longragged3col` style is like `longragged` but with 3 columns

```
\newglossarystyle{longragged3col}{%
```

Use a `longtable` with 3 columns:

```
\renewenvironment{theglossary}{%
  {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}%
    >{\raggedright}p{\glspagelistwidth}}}%
  {\end{longtable}}%
```

No table header:

```
\renewcommand*{\glossaryheader}{}%
```

No headings between groups:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} & ##2\tabularnewline
}%
```

Sub-entries on a separate row (no name, description in second column, page list in third column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
```



```

        \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
    \fi
}

```

`longragged3colborder` The `longragged3colborder` style is like the `longragged3col` style but with a border:

```
\newglossarystyle{longragged3colborder}{%
```

Base it on the `glostylelongragged3col` style:

```
\setglossarystyle{longragged3col}%
```

Use a `longtable` with 3 columns with vertical lines around them:

```

\renewenvironment{theglossary}%
  {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|%
   >{\raggedright}p{\glspagelistwidth}|}}%
  {\end{longtable}}%

```

Place horizontal lines at the head and foot of the table:

```

\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}

```

`longragged3colheader` The `longragged3colheader` style is like `longragged3col` but with a header row:

```
\newglossarystyle{longragged3colheader}{%
```

Base it on the `glostylelongragged3col` style:

```
\setglossarystyle{longragged3col}%
```

Set the table's header:

```

\renewcommand*{\glossaryheader}{%
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries\pagelistname\tabularnewline\endhead}%
}

```

`longragged3colheaderborder` The `longragged3colheaderborder` style is like the above but with a border

```
\newglossarystyle{longragged3colheaderborder}{%
```

Base it on the `glostylelongragged3colborder` style:

```
\setglossarystyle{longragged3colborder}%
```

Set the table's header and add horizontal line at table's foot:

```

\renewcommand*{\glossaryheader}{%
  \hline
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries\pagelistname\tabularnewline\hline\endhead
  \hline\endfoot}%
}

```

`altlongragged4col` The `altlongragged4col` style is like the `altlong4col` style defined in the `glossary-long` package, except that ragged right formatting is used for the description and page list columns.

```
\newglossarystyle{altlongragged4col}{%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
  {\begin{longtable}[1>{\raggedright}p{\glsdescwidth}1%
   >{\raggedright}p{\glspagelistwidth}}}%
  {\end{longtable}}%
```

No table header:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a single row (name in first column, description in second column, symbol in third column, page list in last column):

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} & \glossentrysymbol{##1} &
  ##2\tabularnewline
}%
```

Sub entries on a single row with no name (description in second column, symbol in third column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  \glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
\fi
}
```

`altlongragged4colheader` The `altlongragged4colheader` style is like `altlongragged4col` but with a header row.

```
\newglossarystyle{altlongragged4colheader}{%
```

Base it on the `glostylealtlongragged4col` style:

```
\setglossarystyle{altlongragged4col}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
  {\begin{longtable}[1>{\raggedright}p{\glsdescwidth}1%
   >{\raggedright}p{\glspagelistwidth}}}%
  {\end{longtable}}%
```

Table has a header:

```
\renewcommand*\glossaryheader}{%
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\endhead}%
}
```

`altlongragged4colborder` The `altlongragged4colborder` style is like `altlongragged4col` but with a border.

```
\newglossarystyle{altlongragged4colborder}{%
```

Base it on the `altlongragged4col` style:

```
\setglossarystyle{altlongragged4col}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|}
   >{\raggedright}p{\glspagelistwidth}|}}%
  {\end{longtable}}%
```

Add horizontal lines to the head and foot of the table:

```
\renewcommand*\glossaryheader}{\hline\endhead\hline\endfoot}%
}
```

`altlongragged4colheaderborder` The `altlongragged4colheaderborder` style is like the above but with a header as well as a border.

```
\newglossarystyle{altlongragged4colheaderborder}{%
```

Base it on the `altlongragged4col` style:

```
\setglossarystyle{altlongragged4col}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|}
   >{\raggedright}p{\glspagelistwidth}|}}%
  {\end{longtable}}%
```

Add table header and horizontal line at the table's foot:

```
\renewcommand*\glossaryheader}{%
  \hline\bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\hline\endhead
  \hline\endfoot}%
}
```

3.7 Glossary Styles using `multicol` (`glossary-mcols.sty`)

The `glossary-mcols` style file defines glossary styles that use the `multicol` package. These use the tree-like glossary styles in a `multicol` environment.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-mcols-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-mcols-2021-11-01.sty}
\DeclareCurrentRelease{v4.56}{2025-03-03}
```

Declare package:

```
\ProvidesPackage{glossary-mcols}[2025/03/03 v4.56 (NLCT)]
```

Required packages:

```
\RequirePackage{multicol}
\RequirePackage{glossary-tree}
```

`\indexspace` There are a few classes that don't define `\indexspace`, so provide a definition if it hasn't been defined.

```
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
```

`\glsmcols` Define macro in which to store the number of columns. (Defaults to 2.)

```
\newcommand*\glsmcols{2}
```

`mcolindex` Multi-column index style. Same as the `index`, but puts the glossary in multiple columns. (Ideally the glossary title should go in the optional argument of `multicols`, but the title isn't part of the glossary style.)

```
\newglossarystyle{mcolindex}{%
  \setglossarystyle{index}%
  \renewenvironment{theglossary}%
    {%
      \begin{multicols}{\glsmcols}
      \setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%

      \let\item\glstreeitem
      \let\subitem\glstreesubitem
      \let\subsubitem\glstreesubsubitem
    }%
    {\end{multicols}}%
}
```

`mcolindexgroup` As `mcolindex` but has headings:

```
\newglossarystyle{mcolindexgroup}{%
  \setglossarystyle{mcolindex}%
  \renewcommand*\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt{\glsgrouptitle{##1}}\indexspace}%
}
```

`mcolindexhypergroup` The `mcolindexhypergroup` style is like the `mcolindexgroup` style but has hyper navigation.

```
\newglossarystyle{mcolindexhypergroup}{%
```

Base it on the `glostylemcolindex` style:

```
\setglossarystyle{mcolindex}%
```

Put navigation links to the groups at the start of the glossary:

```
\renewcommand*{\glossaryheader}{%  
  \item\glstreenavigationfmt{\glsnavigation}\indexspace}%
```

Add a heading for each group (with a target). The group's title is in bold followed by a vertical gap.

```
\renewcommand*{\glsgroupheading}[1]{%  
  \item\glstreegroupheaderfmt  
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%  
  \indexspace}%  
}
```

`mcolindexspannav` Similar to `mcolindexhypergroup`, but puts the navigation line in the optional argument of `multicols`.

```
\newglossarystyle{mcolindexspannav}{%  
  \setglossarystyle{index}%  
  \renewenvironment{theglossary}%  
  {%  
  
    \begin{multicols}{\glsncols}[\noindent\glstreenavigationfmt{\glsnavigation}]  
    \setlength{\parindent}{0pt}%  
    \setlength{\parskip}{0pt plus 0.3pt}%  
  
    \let\item\glstreeitem}%  
  {\end{multicols}}%
```

Add a heading for each group (with a target). The group's title is in bold followed by a vertical gap.

```
\renewcommand*{\glsgroupheading}[1]{%  
  \item\glstreegroupheaderfmt  
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%  
  \indexspace}%  
}
```

`mcoltree` Multi-column index style. Same as the tree, but puts the glossary in multiple columns.

```
\newglossarystyle{mcoltree}{%  
  \setglossarystyle{tree}%  
  \renewenvironment{theglossary}%  
  {%  
  
    \begin{multicols}{\glsncols}  
    \setlength{\parindent}{0pt}%  
    \setlength{\parskip}{0pt plus 0.3pt}%  
  }%  
  {\end{multicols}}%  
}
```

`mcoltreegroup` Like the `mcoltree` style but the glossary groups have headings.

```
\newglossarystyle{mcoltreegroup}{%
Base it on the glostylemcoltree style:
\setglossarystyle{mcoltree}%
Each group has a heading (in bold) followed by a vertical gap):
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
```

`mcoltreehypergroup` The `mcoltreehypergroup` style is like the `treegroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{mcoltreehypergroup}{%
Base it on the glostylemcoltree style:
\setglossarystyle{mcoltree}%
Put navigation links to the groups at the start of the theglossary environment:
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
Each group has a heading (in bold with a target) followed by a vertical gap):
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
```

`mcoltreesspannav` Similar to the `mcoltreehypergroup` style but the navigation line is put in the optional argument of the `multicols` environment.

```
\newglossarystyle{mcoltreesspannav}{%
\setglossarystyle{tree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsncols}[ \noindent\glstreenavigationfmt{\glsnavigation}]
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%
```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
```

`mcoltreenoname` Multi-column index style. Same as the `treenoname`, but puts the glossary in multiple columns.

```

\newglossarystyle{mcoltreename}{%
  \setglossarystyle{treename}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}

```

`mcoltreenamegroup` Like the `mcoltreename` style but the glossary groups have headings.

```

\newglossarystyle{mcoltreenamegroup}{%
Base it on the glostylemcoltreename style:
  \setglossarystyle{mcoltreename}%
Give each group a heading:
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}

```

`mcoltreenamehypergroup` The `mcoltreenamehypergroup` style is like the `mcoltreenamegroup` style, but has a set of links to the groups at the start of the glossary.

```

\newglossarystyle{mcoltreenamehypergroup}{%
Base it on the glostylemcoltreename style:
  \setglossarystyle{mcoltreename}%
Put navigation links to the groups at the start of the theglossary environment:
  \renewcommand*\glossaryheader{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
Each group has a heading (in bold with a target) followed by a vertical gap):
  \renewcommand*\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}

```

`mcoltreenamepannav` Similar to the `mcoltreenamehypergroup` style but the navigation line is put in the optional argument of the `multicols` environment.

```

\newglossarystyle{mcoltreenamepannav}{%
  \setglossarystyle{treename}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}

```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupheading}[1]{%
  \par\noindent
  \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}
```

`mcolalmtree` Multi-column index style. Same as the `almtree`, but puts the glossary in multiple columns.

```
\newglossarystyle{mcolalmtree}{%
  \setglossarystyle{almtree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
}
```

`mcolalmtreegroup` Like the `mcolalmtree` style but the glossary groups have headings.

```
\newglossarystyle{mcolalmtreegroup}{%
```

Base it on the `glostylemcolalmtree` style:

```
\setglossarystyle{mcolalmtree}%
```

Give each group a heading.

```
\renewcommand{\glsgroupheading}[1]{\par
  \def\@gls@prevlevel{-1}%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
```

`mcolalmtreehypergroup` The `mcolalmtreehypergroup` style is like the `mcolalmtreegroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{mcolalmtreehypergroup}{%
```

Base it on the `glostylemcolalmtree` style:

```
\setglossarystyle{mcolalmtree}%
```

Put the navigation links in the header

```
\renewcommand*{\glossaryheader}{%
  \par
  \def\@gls@prevlevel{-1}%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstreenavigationfmt{\glsnavigation}\par\indexspace}%
}
```


Put a hypertarget at the start of each group

```
\renewcommand*{\glsgroupheading}[1]{%
  \par
  \def\@gls@prevlevel{-1}%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}
```

`mcolalttreespannav` Similar to the `mcolalttreehypergroup` style but the navigation line is put in the optional argument of the `multicols` environment.

```
\newglossarystyle{mcolalttreespannav}{%
  \setglossarystyle{alttree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glscols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
```

Put a hypertarget at the start of each group

```
\renewcommand*{\glsgroupheading}[1]{%
  \par
  \def\@gls@prevlevel{-1}%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}
```

3.8 Glossary Styles using supertabular environment (`glossary-super.sty`)

The glossary styles defined in the `glossary-super` package use the `supertabular` environment.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-super-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-super-2021-11-01.sty}
\DeclareCurrentRelease{v4.56}{2025-03-03}
```

Declare package:

```
\ProvidesPackage{glossary-super}[2025/03/03 v4.56 (NLCT)]
```

Requires the `supertabular` package:

```
\RequirePackage{supertabular}
```

`\glsdescwidth` This is a length that governs the width of the description column. This may already have been defined if `glossary-long` has been loaded.

```
\@ifundefined{glsdescwidth}{%
  \newlength\glsdescwidth
  \setlength{glsdescwidth}{0.6\hsize}
}
```

`\glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined if `glossary-long` has been loaded.

```
\@ifundefined{glspagelistwidth}{%
  \newlength\glspagelistwidth
  \setlength{glspagelistwidth}{0.1\hsize}
}
```

`super` The super glossary style uses the `supertabular` environment (it uses lengths defined in the `glossary-long` package.)

```
\newglossarystyle{super}{%
```

Put the glossary in a `supertabular` environment with two columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{}}\tabletail{}%
  \begin{supertabular}{lp{glsdescwidth}}%
  {\end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries put in a row (name in first column, description and page list in second column):

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
}%
```

Sub entries put in a row (no name, description and page list in second column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
  ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
```

```

        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{& \tabularnewline}%
    \fi
}

```

superborder The superborder style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{superborder}{%
```

Base it on the `glostylesuper` style:

```
\setglossarystyle{super}%
```

Put the glossary in a `supertabular` environment with two columns and a horizontal line in the head and tail:

```

\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  \begin{supertabular}{|l|p{\glsdescwidth}|}%
  {\end{supertabular}}%
}

```

superheader The superheader style is like the `super` style, but with a header:

```
\newglossarystyle{superheader}{%
```

Base it on the `glostylesuper` style:

```
\setglossarystyle{super}%
```

Put the glossary in a `supertabular` environment with two columns, a header and no tail:

```

\renewenvironment{theglossary}%
  {\tablehead{\bfseries \entryname &
  \bfseries \descriptionname \tabularnewline}%
  \tabletail{}}%
  \begin{supertabular}{lp{\glsdescwidth}}%
  {\end{supertabular}}%
}

```

superheaderborder The superheaderborder style is like the `super` style but with a header and border:

```
\newglossarystyle{superheaderborder}{%
```

Base it on the `glostylesuper` style:

```
\setglossarystyle{super}%
```

Put the glossary in a `supertabular` environment with two columns, a header and horizontal lines above and below the table:

```

\renewenvironment{theglossary}%
  {\tablehead{\hline\bfseries \entryname &
  \bfseries \descriptionname \tabularnewline \hline}%
  \tabletail{\hline}
  \begin{supertabular}{|l|p{\glsdescwidth}|}%
  {\end{supertabular}}%
}

```

`super3col` The `super3col` style is like the `super` style, but with 3 columns:

```
\newglossarystyle{super3col}{%
```

Put the glossary in a `supertabular` environment with three columns and no head or tail:

```
\renewenvironment{theglossary}%  
  {\tablehead{ }\tabletail{ }%  
   \begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}%  
   {\end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgrupheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
\renewcommand{\glossentry}[2]{%  
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &  
  \glossentrydesc{##1} & ##2\tabularnewline  
}%
```

Sub entries on a row (no name, description in second column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%  
  &  
  \glssubentryitem{##2}%  
  \glstarget{##2}{\strut}\glossentrydesc{##2} &  
  ##3\tabularnewline  
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip  
  \renewcommand*{\glsgroupskip}{}%  
\else  
  \renewcommand*{\glsgroupskip}{& & \tabularnewline}%  
\fi  
}
```

`super3colborder` The `super3colborder` style is like the `super3col` style, but with a border:

```
\newglossarystyle{super3colborder}{%
```

Base it on the `glostylesuper3col` style:

```
\setglossarystyle{super3col}%
```

Put the glossary in a `supertabular` environment with three columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%  
  {\tablehead{\hline}\tabletail{\hline}%
```

```

        \begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
        {\end{supertabular}}%
    }

```

super3colheader The `super3colheader` style is like the `super3col` style but with a header row:

```

\newglossarystyle{super3colheader}{%
Base it on the glostylesuper3col style:
    \setglossarystyle{super3col}%
Put the glossary in a supertabular environment with three columns, a header
and no tail:
    \renewenvironment{theglossary}%
        {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
            \bfseries\pagelistname\tabularnewline}\tabletail{}}%
        \begin{supertabular}{|p{\glsdescwidth}p{\glspagelistwidth}|}%
        {\end{supertabular}}%
    }

```

super3colheaderborder The `super3colheaderborder` style is like the `super3col` style but with a header and border:

```

\newglossarystyle{super3colheaderborder}{%
Base it on the glostylesuper3colborder style:
    \setglossarystyle{super3colborder}%
Put the glossary in a supertabular environment with three columns, a header
with horizontal lines and a horizontal line in the tail:
    \renewenvironment{theglossary}%
        {\tablehead{\hline
            \bfseries\entryname&\bfseries\descriptionname&
            \bfseries\pagelistname\tabularnewline\hline}%
        \tabletail{\hline}%
        \begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
        {\end{supertabular}}%
    }

```

super4col The `super4col` glossary style has four columns, where the third column contains the value of the corresponding symbol key used when that entry was defined.

```

\newglossarystyle{super4col}{%
Put the glossary in a supertabular environment with four columns and no head
or tail:
    \renewenvironment{theglossary}%
        {\tablehead{} \tabletail{}}%
        \begin{supertabular}{|l|l|l|l|}%
        \end{supertabular}}%
Do nothing at the start of the table:
    \renewcommand*{\glossaryheader}{}%

```

No group headings:

```
\renewcommand*\glsgroupheading}[1]{}
```

Main (level 0) entries on a row with the name in the first column, description in second column, symbol in third column and page list in last column:

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  \glossentrysymbol{##1} & ##2\tabularnewline
}%
```

Sub entries on a row with no name, the description in the second column, symbol in third column and page list in last column:

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  \glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip}{& & \tabularnewline}%
\fi
}
```

`super4colheader` The `super4colheader` style is like the `super4col` but with a header row.

```
\newglossarystyle{super4colheader}{%
```

Base it on the `glostylesuper4col` style:

```
\setglossarystyle{super4col}%
```

Put the glossary in a `supertabular` environment with four columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
    \bfseries\symbolname &
    \bfseries\pagelistname\tabularnewline}%
  \tabletail{}%
  \begin{supertabular}{l111}}%
  {\end{supertabular}}%
}
```

`super4colborder` The `super4colborder` style is like the `super4col` but with a border.

```
\newglossarystyle{super4colborder}{%
```

Base it on the `glostylesuper4col` style:

```
\setglossarystyle{super4col}%
```

Put the glossary in a `supertabular` environment with four columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  \begin{supertabular}{|1|1|1|1|}%
  {\end{supertabular}}%
}
```

`super4colheaderborder` The `super4colheaderborder` style is like the `super4col` but with a header and border.

```
\newglossarystyle{super4colheaderborder}{%
```

Base it on the `glostylessuper4col` style:

```
\setglossarystyle{super4col}%
```

Put the glossary in a `supertabular` environment with four columns and a header bordered by horizontal lines and a horizontal line in the tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline\bfseries\entryname&\bfseries\descriptionname&
  \bfseries\symbolname &
  \bfseries\pagelistname\tabularnewline\hline}%
  \tabletail{\hline}%
  \begin{supertabular}{|1|1|1|1|}%
  {\end{supertabular}}%
}
```

`altsuper4col` The `altsuper4col` glossary style is like `super4col` but has provision for multiline descriptions.

```
\newglossarystyle{altsuper4col}{%
```

Base it on the `glostylessuper4col` style:

```
\setglossarystyle{super4col}%
```

Put the glossary in a `supertabular` environment with four columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{}\tabletail{}%
  \begin{supertabular}{lp{\glsdescwidth}lp{\glspagelistwidth}}%
  {\end{supertabular}}%
}
```

`altsuper4colheader` The `altsuper4colheader` style is like the `altsuper4col` but with a header row.

```
\newglossarystyle{altsuper4colheader}{%
```

Base it on the `glostylessuper4colheader` style:

```
\setglossarystyle{super4colheader}%
```

Put the glossary in a `supertabular` environment with four columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
```

```

        \bfseries\symbolname &
        \bfseries\pagelistname\tabularnewline\tabletail{}%
        \begin{supertabular}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
        {\end{supertabular}}%
    }

```

altsuper4colborder The `altsuper4colborder` style is like the `altsuper4col` but with a border.

```
\newglossarystyle{altsuper4colborder}{%
```

Base it on the `glostylesuper4colborder` style:

```
\setglossarystyle{super4colborder}%
```

Put the glossary in a `supertabular` environment with four columns and a horizontal line in the head and tail:

```

\renewenvironment{theglossary}%
{
\tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}%
{
|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%
{\end{supertabular}}%
}

```

altsuper4colheaderborder The `altsuper4colheaderborder` style is like the `altsuper4col` but with a header and border.

```
\newglossarystyle{altsuper4colheaderborder}{%
```

Base it on the `glostylesuper4colheaderborder` style:

```
\setglossarystyle{super4colheaderborder}%
```

Put the glossary in a `supertabular` environment with four columns and a header bordered by horizontal lines and a horizontal line in the tail:

```

\renewenvironment{theglossary}%
{
\tablehead{\hline
\bfseries\entryname &
\bfseries\descriptionname &
\bfseries\symbolname &
\bfseries\pagelistname\tabularnewline\hline}%
\tabletail{\hline}%
\begin{supertabular}%
{
|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%
{\end{supertabular}}%
}

```

3.9 Glossary Styles using `supertabular` environment (`glossary-superragged.sty`)

The glossary styles defined in the `glossary-superragged` package use the `supertabular` environment. These styles are like those provided by the `glossary-super` package, except that the multiline columns have ragged right justification.

```
\NeedsTeXFormat{LaTeX2e}
```


Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-superragged-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-superragged-2021-11-01.sty}
\DeclareCurrentRelease{v4.56}{2025-03-03}
```

Declare package:

```
\ProvidesPackage{glossary-superragged}[2025/03/03 v4.56 (NLCT)]
```

Requires the array package:

```
\RequirePackage{array}
```

Requires the supertabular package:

```
\RequirePackage{supertabular}
```

`\glsdescwidth` This is a length that governs the width of the description column. This may already have been defined.

```
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{}
```

`\glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined.

```
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{}
```

`superragged` The superragged glossary style uses the supertabular environment.

```
\newglossarystyle{superragged}{%
```

Put the glossary in a supertabular environment with two columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{}\tabletail{}%
  \begin{supertabular}{1>{\raggedright}p{\glsdescwidth}}%
  {\end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries put in a row (name in first column, description and page list in second column):

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1}\glspostdescription\space ##2%
  \tabularnewline
}%
```

Sub entries put in a row (no name, description and page list in second column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
  ##3%
  \tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& \tabularnewline}%
\fi
}
```

`superraggedborder` The `superraggedborder` style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{superraggedborder}{%
```

Base it on the `glostylesuperragged` style:

```
\setglossarystyle{superragged}%
```

Put the glossary in a `supertabular` environment with two columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|}%
  {\end{supertabular}}%
}
```

`superraggedheader` The `superraggedheader` style is like the `super` style, but with a header:

```
\newglossarystyle{superraggedheader}{%
```

Base it on the `glostylesuperragged` style:

```
\setglossarystyle{superragged}%
```

Put the glossary in a `supertabular` environment with two columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries \entryname & \bfseries \descriptionname
  \tabularnewline}%
  \tabletail{}%
  \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}}%
  {\end{supertabular}}%
}
```

`superraggedheaderborder` The `superraggedheaderborder` style is like the `superragged` style but with a header and border:

```
\newglossarystyle{superraggedheaderborder}{%
```

Base it on the `glostylesuper` style:

```
\setglossarystyle{superragged}%
```

Put the glossary in a `supertabular` environment with two columns, a header and horizontal lines above and below the table:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline\bfseries \entryname &
    \bfseries \descriptionname\tabularnewline\hline}%
  \tabletail{\hline}
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|}%
  \end{supertabular}}%
}
```

`superragged3col` The `superragged3col` style is like the `superragged` style, but with 3 columns:

```
\newglossarystyle{superragged3col}{%
```

Put the glossary in a `supertabular` environment with three columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{} \tabletail{}%
  \begin{supertabular}{|l>{\raggedright}p{\glsdescwidth}%
    >{\raggedright}p{\glspagelistwidth}}%
  \end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  ##2\tabularnewline
}%
```

Sub entries on a row (no name, description in second column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip{ }%
\else
  \renewcommand*\glsgroupskip{& & \tabularnewline}%
\fi
}
```

`superragged3colborder` The `superragged3colborder` style is like the `superragged3col` style, but with a border:

```
\newglossarystyle{superragged3colborder}{%
```

Base it on the `glostylesuperragged3col` style:

```
\setglossarystyle{superragged3col}%
```

Put the glossary in a `supertabular` environment with three columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
  >{\raggedright}p{\glspagelistwidth}|}%
  {\end{supertabular}}%
}
```

`superragged3colheader` The `superragged3colheader` style is like the `superragged3col` style but with a header row:

```
\newglossarystyle{superragged3colheader}{%
```

Base it on the `glostylesuperragged3col` style:

```
\setglossarystyle{superragged3col}%
```

Put the glossary in a `supertabular` environment with three columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
  \bfseries\pagelistname\tabularnewline}\tabletail{ }%
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}%
  >{\raggedright}p{\glspagelistwidth}}%
  {\end{supertabular}}%
}
```

`raggedright3colheaderborder` The `superragged3colheaderborder` style is like the `superragged3col` style but with a header and border:

```
\newglossarystyle{superragged3colheaderborder}{%
```

Base it on the `glostylesuperragged3colborder` style:

```
\setglossarystyle{superragged3colborder}%
```

Put the glossary in a supertabular environment with three columns, a header with horizontal lines and a horizontal line in the tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline}%
  \tabletail{\hline}%
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
    >{\raggedright}p{\glspagelistwidth}|}%
  \end{supertabular}}%
}
```

`altsuperragged4col` The `altsuperragged4col` glossary style is like `altsuper4col` style in the `glossary-super` package but uses ragged right formatting in the description and page list columns.

```
\newglossarystyle{altsuperragged4col}{%
```

Put the glossary in a supertabular environment with four columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{}\tabletail{}%
  \begin{supertabular}{|l>{\raggedright}p{\glsdescwidth}l%
    >{\raggedright}p{\glspagelistwidth}|}%
  \end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row with the name in the first column, description in second column, symbol in third column and page list in last column:

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  \glossentrysymbol{##1} & ##2\tabularnewline
}%
```

Sub entries on a row with no name, the description in the second column, symbol in third column and page list in last column:

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  \glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```

\ifglsgroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi
}

```

`altsuperragged4colheader` The `altsuperragged4colheader` style is like the `altsuperragged4col` style but with a header row.

```
\newglossarystyle{altsuperragged4colheader}{%
```

Base it on the `glostylealtsuperragged4col` style:

```
\setglossarystyle{altsuperragged4col}%
```

Put the glossary in a `supertabular` environment with four columns, a header and no tail:

```

\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
  \bfseries\symbolname &
  \bfseries\pagelistname\tablearnewline}\tabletail{}}%
  \begin{supertabular}{1>{\raggedright}p{\glsgdescwidth}1%
  >{\raggedright}p{\glspagelistwidth}}}%
  {\end{supertabular}}%
}

```

`altsuperragged4colborder` The `altsuperragged4colborder` style is like the `altsuperragged4col` style but with a border.

```
\newglossarystyle{altsuperragged4colborder}{%
```

Base it on the `glostylealtsuperragged4col` style:

```
\setglossarystyle{altsuper4col}%
```

Put the glossary in a `supertabular` environment with four columns and a horizontal line in the head and tail:

```

\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  \begin{supertabular}%
  {1|>{\raggedright}p{\glsgdescwidth}1|1%
  >{\raggedright}p{\glspagelistwidth}|}}%
  {\end{supertabular}}%
}

```

`superragged4colheaderborder` The `altsuperragged4colheaderborder` style is like the `altsuperragged4col` style but with a header and border.

```
\newglossarystyle{altsuperragged4colheaderborder}{%
```

Base it on the `glostylealtsuperragged4col` style:

```
\setglossarystyle{altsuperragged4col}%
```

Put the glossary in a `supertabular` environment with four columns and a header bordered by horizontal lines and a horizontal line in the tail:

```

\renewenvironment{theglossary}%
  {\tablehead{\hline
    \bfseries\entryname &
    \bfseries\descriptionname &
    \bfseries\symbolname &
    \bfseries\pagelistname\tabularnewline\hline}%
  \tabletail{\hline}%
  \begin{supertabular}%
    {||>{\raggedright}p{\glsdescwidth}||%
    >{\raggedright}p{\glspagelistwidth}||}%
  {\end{supertabular}}%
}

```

3.10 Tree Styles (`glossary-tree.sty`)

The `glossary-tree` style file defines glossary styles that have a tree-like structure. These are designed for hierarchical glossaries.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```

\DeclareRelease{v4.46}{2020-03-19}{glossary-tree-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-tree-2021-11-01.sty}
\DeclareCurrentRelease{v4.56}{2025-03-03}

```

Declare package:

```
\ProvidesPackage{glossary-tree}[2025/03/03 v4.56 (NLCT)]
```

`\indexspace` There are a few classes that don't define `\indexspace`, so provide a definition if it hasn't been defined.

```

\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}

```

`\glstreenamefmt` Format used to display the name in the tree styles. (This may be counteracted by `\glsnamefont`.) This command was previously also used to format the group headings.

```
\newcommand*{\glstreenamefmt}[1]{\textbf{#1}}
```

`\glstreegroupheaderfmt` Format used to display the group header in the tree styles. Before v4.22, `\glstreenamefmt` was used for the group header, so the default definition uses that to help maintain backward-compatibility, since in previous versions redefining `\glstreenamefmt` would've also affected the group headings.

```
\newcommand*{\glstreegroupheaderfmt}[1]{\glstreenamefmt{#1}}
```

`\glstreenavigationfmt` Format used to display the navigation header in the tree styles.

```
\newcommand*{\glstreenavigationfmt}[1]{\glstreenamefmt{#1}}
```

Allow the user to adjust the index style without disturbing the index.

`\glstreeitem` Top level item used in index style.

```
\ifdef\@idxitem
{\newcommand{\glstreeitem}{\@idxitem}}
{\newcommand{\glstreeitem}{\par\hangindent40\p@}}
```

`\glstreesubitem` Level 1 item used in index style.

```
\ifdef\subitem
{\let\glstreesubitem\subitem}
{\newcommand\glstreesubitem{\glstreeitem\hspace*{20\p@}}}
```

`\glstreesubsubitem` Level 1 item used in index style.

```
\ifdef\subsubitem
{\let\glstreesubsubitem\subsubitem}
{\newcommand\glstreesubsubitem{\glstreeitem\hspace*{30\p@}}}
```

`\glstreepredesc` Allow the user to adjust the space before the description (except for the `almtree` style).

```
\newcommand{\glstreepredesc}{\space}
```

`\glstreechildpredesc` Allow the user to adjust the space before the description for sub-entries (except for the `treenoname` and `almtree` style).

```
\newcommand{\glstreechildpredesc}{\space}
```

`index` The index glossary style is similar in style to the way indices are usually typeset using `\item`, `\subitem` and `\subsubitem`. The entry name is set in bold. If an entry has a symbol, it is placed in brackets after the name. Then the description is displayed, followed by the number list. This style allows up to three levels.

```
\newglossarystyle{index}{%
```

Set the paragraph indentation and skip and define `\item` to be the same as that used by `theindex`:

```
\renewenvironment{theglossary}%
{\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
\let\item\glstreeitem
\let\subitem\glstreesubitem
\let\subsubitem\glstreesubsubitem
}%
{\par}%
```

Do nothing at the start of the environment:

```
\renewcommand*{\glossaryheader}{}%
```

No group headers:

```
\renewcommand*{\glsgroupheading}[1]{}%
```


Main (level 0) entry starts a new item with the name in bold followed by the symbol in brackets (if it exists), the description and the page list.

```
\renewcommand*{\glossentry}[2]{%
  \item\glstreeitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
  \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
  \glstreepredesc \glossentrydesc{##1}\glspostdescription\space ##2%
}%
```

Sub entries: level 1 entries use `\subitem`, levels greater than 1 use `\subsubitem`. The level (`##1`) shouldn't be 0, as that's catered by `\glossentry`, but for completeness, if the level is 0, `\item` is used. The name is put in bold, followed by the symbol in brackets (if it exists), the description and the page list.

```
\renewcommand{\subglossentry}[3]{%
  \ifcase##1\relax
    % level 0
    \item
  \or
    % level 1
    \subitem
    \glssubentryitem{##2}%
  \else
    % all other levels
    \subsubitem
  \fi
  \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
  \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
  \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3%
}%
```

Vertical gap between groups is the same as that used by indices:

```
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
```

`indexgroup` The `indexgroup` style is like the `index` style but has headings.

```
\newglossarystyle{indexgroup}{%
```

Base it on the `glostyleindex` style:

```
\setglossarystyle{index}%
```

Add a heading for each group. This puts the group's title in bold followed by a vertical gap.

```
\renewcommand*{\glsgroupheading}[1]{%
  \item\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
  \indexspace
}%
}
```

`indexhypergroup` The `indexhypergroup` style is like the `indexgroup` style but has hyper navigation.

```
\newglossarystyle{indexhypergroup}{%
```

Base it on the `glostyleindex` style:

```
\setglossarystyle{index}%
```

Put navigation links to the groups at the start of the glossary:

```
\renewcommand*\glossaryheader}{%
  \item\glstreenavigationfmt{\glsnavigation}\indexspace}%
```

Add a heading for each group (with a target). The group's title is in bold followed by a vertical gap.

```
\renewcommand*\glsgroupheading}[1]{%
  \item\glstreegroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
  \indexspace}%
}
```

tree The tree glossary style is similar in style to the index style, but can have arbitrary levels.

```
\newglossarystyle{tree}{%
```

Set the paragraph indentation and skip:

```
\renewenvironment{theglossary}%
  {\setlength{\parindent}{0pt}%
   \setlength{\parskip}{0pt plus 0.3pt}}%
  {}%
```

Do nothing at the start of the theglossary environment:

```
\renewcommand*\glossaryheader}{}%
```

No group headings:

```
\renewcommand*\glsgroupheading}[1]{}%
```

Main (level 0) entries: name in bold, followed by symbol in brackets (if it exists), the description and the page list:

```
\renewcommand{\glossentry}[2]{%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
  \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
  \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
}%
```

Sub entries: level $\langle n \rangle$ is indented by $\langle n \rangle$ times `\glstreeindent`. The name is in bold, followed by the symbol in brackets (if it exists), the description and the page list.

```
\renewcommand{\subglossentry}[3]{%
  \hangindent##1\glstreeindent\relax
  \parindent##1\glstreeindent\relax
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
  \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
  \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3\par
}%
```

Vertical gap between groups is the same as that used by indices:

```
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
```

treegroup Like the tree style but the glossary groups have headings.

```
\newglossarystyle{treegroup}{%
```

Base it on the glostyletree style:

```
\setglossarystyle{tree}%
```

Each group has a heading (in bold) followed by a vertical gap):

```
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par
\indexspace}%
}
```

treehypergroup The treehypergroup style is like the treegroup style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{treehypergroup}{%
```

Base it on the glostyletree style:

```
\setglossarystyle{tree}%
```

Put navigation links to the groups at the start of the theglossary environment:

```
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
```

\glstreeindent Length governing left indent for each level of the tree style.

```
\newlength\glstreeindent
\setlength{\glstreeindent}{10pt}
```

treenoname The treenoname glossary style is like the tree style, but doesn't print the name or symbol for sub-levels.

```
\newglossarystyle{treenoname}{%
```

Set the paragraph indentation and skip:

```
\renewenvironment{theglossary}%
{\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}}%
{}
```

No header:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*\glsgroupheading}[1]{}
```

Main (level 0) entries: the name is in bold, followed by the symbol in brackets (if it exists), the description and the page list.

```
\renewcommand{\glossentry}[2]{%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
  \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
  \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
}%
```

Sub entries: level $\langle n \rangle$ is indented by $\langle n \rangle$ times `\glstreeindent`. The name and symbol are omitted. The description followed by the page list are displayed.

```
\renewcommand{\subglossentry}[3]{%
  \hangindent##1\glstreeindent\relax
  \parindent##1\glstreeindent\relax
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \glstarget{##2}{\strut}%
  \glossentrydesc{##2}\glspostdescription\space##3\par
}%
```

Vertical gap between groups is the same as that used by indices:

```
\renewcommand*\glsgroupskip{\ifglsnogroupskip\else\indexspace\fi}%
}
```

`treenonamegroup` Like the `treenoname` style but the glossary groups have headings.

```
\newglossarystyle{treenonamegroup}{%
```

Base it on the `glostyletreenoname` style:

```
\setglossarystyle{treenoname}%
```

Give each group a heading:

```
\renewcommand{\glsgroupheading}[1]{\par
  \noindent\glstreegroupheaderfmt
  {\glsgetgrouptitle{##1}}\par\indexspace}%
}
```

`treenonamehypergroup` The `treenonamehypergroup` style is like the `treenonamegroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{treenonamehypergroup}{%
```

Base it on the `glostyletreenoname` style:

```
\setglossarystyle{treenoname}%
```

Put navigation links to the groups at the start of the `theglossary` environment:

```
\renewcommand*\glossaryheader{%
  \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupeheading}[1]{%
  \par\noindent
  \glstreenameheaderfmt
  {\glshnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
  \indexspace}%
}
```

`\glswidestname` Find the widest name over all parentless entries in the given glossary or glossaries.

```
\newrobustcmd*{\glswidestname}[1][\glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  \forallglossaries[#1]{\gls@type}%
  {%
    \forallglsentries[\gls@type]{\glo@label}%
    {%
      \ifglshasparent{\glo@label}%
      {}%
      {%
        \glsmeasurewidth{\dimen@}%
        {\glstreenamefmt{\glsentryname{\glo@label}}}%
        \ifdim\dimen@>\gls@tmplen
          \gls@tmplen=\dimen@
          \letcs{\glswidestname}{glo@\glsdetoklabel{\glo@label}@name}%
        \fi
      }%
    }%
  }%
}
```

`\glsssetwidest` `\glsssetwidest[level]{text}` sets the widest text for the given level. It is used by the `alttree` glossary styles to determine the indentation of each level.

```
\newcommand*{\glsssetwidest}[2][0]{%
  \expandafter\def\csname @glswidestname\romannumeral#1\endcsname{%
  #2}%
}
```

`\@glswidestname` Initialise `\@glswidestname`.

```
\newcommand*{\@glswidestname}{}
```

`\glstreenamebox` Used by the `alttree` style to create the box for the name and associated information.

```
\newcommand*{\glstreenamebox}[2]{%
  \makebox[#1][l]{#2}%
}
```

`alttree` The `alttree` glossary style is similar in style to the `tree` style, but the indentation is obtained from the width of `\@glswidestname` which is set using `\glsssetwidest`.

```
\newglossarystyle{alttree}{%
```

Redefine theglossary environment.

```
\renewenvironment{theglossary}%  
  {\def\@gls@prevlevel{-1}%  
   \mbox{}\par}%  
  {\par}%
```

Set the header and group headers to nothing.

```
\renewcommand*\glossaryheader{}%  
\renewcommand*\glsgroupheading}[1]{}%
```

Redefine the way that the level 0 entries are displayed.

```
\renewcommand{\glossentry}[2]{%  
  \ifnum\@gls@prevlevel=0\relax  
  \else
```

Find out how big the indentation should be by measuring the widest entry.

```
  \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%  
  \fi
```

Set the hangindent and paragraph indent.

```
  \hangindent\glstreeindent  
  \parindent\glstreeindent
```

Put the name to the left of the paragraph block.

```
  \makebox[0pt][r]{\glstreenamebox{\glstreeindent}{%  
    \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}}%  
  }
```

If the symbol is missing, ignore it, otherwise put it in brackets.

```
  \ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}%
```

Do the description followed by the description terminator and location list.

```
  \glossentrydesc{##1}\glspostdescription \space ##2\par
```

Set the previous level to 0.

```
  \def\@gls@prevlevel{0}%  
  }%
```

Redefine the way sub-entries are displayed.

```
\renewcommand{\subglossentry}[3]{%
```

Increment and display the sub-entry counter if this is a level 1 entry and the sub-entry counter is in use.

```
  \ifnum##1=1\relax  
    \glssubentryitem{##2}%  
  \fi
```

If the level hasn't changed, keep the same settings, otherwise adjust `\glstreeindent` accordingly.

```
  \ifnum\@gls@prevlevel=##1\relax  
  \else
```

Compute the widest entry for this level, or for level 0 if not defined for this level.

Store in `\gls@tmplen`

```
\@ifundefined{@glswidestname\romannumeral##1}{%
  \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}{%
  \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{%
    \csname @glswidestname\romannumeral##1\endcsname\space}}}%
```

Determine if going up or down a level

```
\ifnum\@gls@prevlevel<##1\relax
```

Depth has increased, so add the width of the widest entry to `\glstreeindent`.

```
\setlength\glstreeindent\gls@tmplen
\addtolength\glstreeindent\parindent
\parindent\glstreeindent
\else
```

Depth has decreased, so subtract width of the widest entry from the previous level to `\glstreeindent`. First determine the width of the widest entry for the previous level and store in `\glstreeindent`.

```
\@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
  \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{%
    \@glswidestname\space}}{%
  \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{%
    \csname @glswidestname\romannumeral\@gls@prevlevel
    \endcsname\space}}}%
```

Subtract this length from the previous level's paragraph indent and set to `\glstreeindent`.

```
\addtolength\parindent{-\glstreeindent}%
\setlength\glstreeindent\parindent
\fi
\fi
```

Set the hanging indentation.

```
\hangindent\glstreeindent
```

Put the name to the left of the paragraph block

```
\makebox[Opt][r]{\glstreenamebox{\gls@tmplen}{%
  \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
```

If the symbol is missing, ignore it, otherwise put it in brackets.

```
\ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
```

Do the description followed by the description terminator and location list.

```
\glossentrydesc{##2}\glspostdescription\space ##3\par
```

Set the previous level macro to the current level.

```
\def\@gls@prevlevel{##1}%
}%
```

Vertical gap between groups is the same as that used by indices:

```
\renewcommand*{\glsgroupskip}{\ifglsgroupskip\else\indexspace\fi}%
}
```

`almtreegroup` Like the `almtree` style but the glossary groups have headings.

```
\newglossarystyle{almtreegroup}{%
Base it on the glostylealmtree style:
\setglossarystyle{almtree}%
Give each group a heading.
\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
\par\indexspace}%
}
```

`almtreehypergroup` The `almtreehypergroup` style is like the `almtreegroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{almtreehypergroup}{%
Base it on the glostylealmtree style:
\setglossarystyle{almtree}%
Put the navigation links in the header
\renewcommand*{\glossaryheader}{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
Put a hypertarget at the start of each group
\renewcommand*{\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}}
```

4 Multi-Lingual Support

4.1 Babel Captions (`glossaries-babel.sty`)

Many thanks to everyone who contributed to the translations both via email and on `comp.text.tex`. Language support has now been split off into independent language modules.

```
\NeedsTeXFormat{LaTeX2e}
```


Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossaries-babel-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-babel-2021-11-01.sty}
\DeclareCurrentRelease{v4.56}{2025-03-03}
```

Declare package

```
\ProvidesPackage{glossaries-babel}[2025/03/03 v4.56 (NLCT)]
```

Load tracklang to obtain language settings.

```
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
```

Check for tracked languages:

```
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
  }%
  \PackageWarningNoLine{glossaries}%
  {No language module detected for ‘\this@dialect’.\MessageBreak
  Language modules need to be installed separately.\MessageBreak
  Please check on CTAN for a bundle called\MessageBreak
  ‘glossaries-\CurrentTrackedLanguage’ or similar}%
}%
}%
{}
```

4.2 Polyglossia Captions (glossaries-polyglossia.sty)

Language support has now been split off into independent language modules.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossaries-polyglossia-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-polyglossia-2021-11-01.sty}
\DeclareCurrentRelease{v4.56}{2025-03-03}
```

Declare package

```
\ProvidesPackage{glossaries-polyglossia}[2025/03/03 v4.56 (NLCT)]
```

Load tracklang to obtain language settings.

```
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
```

Check for tracked languages:

```
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
  }%
  \PackageWarningNoLine{glossaries}%
  {No language module detected for ‘\this@dialect’.\MessageBreak
  Language modules need to be installed separately.\MessageBreak
  Please check on CTAN for a bundle called\MessageBreak
  ‘glossaries-\CurrentTrackedLanguage’ or similar}%
}%
}%
{}
```

5 Prefix Support (glossaries-prefix Code)

This package provides a means of adding prefixes to your glossary entries. For example, you may want to use “a \gls{*label*}” on first use but use “an \gls{*label*}” on subsequent use.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossaries-prefix-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-prefix-2021-11-01.sty}
\DeclareCurrentRelease{v4.56}{2025-03-03}
```

Declare package:

```
\ProvidesPackage{glossaries-prefix}[2025/03/03 v4.56 (NLCT)]
```

Pass all options to glossaries:

```
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
```

Process options:

```
\ProcessOptions
```

Load glossaries:

```
\RequirePackage{glossaries}
```

Add the new keys:

```
\define@key{glossentry}{prefixfirst}{\def\@glo@entryprefixfirst{#1}}%
\define@key{glossentry}{prefixfirstplural}{\def\@glo@entryprefixfirstplural{#1}}%
\define@key{glossentry}{prefix}{\def\@glo@entryprefix{#1}}%
\define@key{glossentry}{prefixplural}{\def\@glo@entryprefixplural{#1}}%
```

Add them to \@gls@keymap:

```
\appto\@gls@keymap{,%  
  {prefixfirst}{prefixfirst},%  
  {prefixfirstplural}{prefixfirstplural},%  
  {prefix}{prefix},%  
  {prefixplural}{prefixplural}%  
}
```

Set the default values:

```
\appto\@newglossaryentryprehook{%  
  \def\@glo@entryprefix{}%  
  \def\@glo@entryprefixplural{}%  
  \let\@glo@entryprefixfirst\@gls@default@value  
  \let\@glo@entryprefixfirstplural\@gls@default@value  
}
```

Set the assignment code:

```
\appto\@newglossaryentryposthook{%  
  \gls@assign@field{}\@glo@label}{prefix}\@glo@entryprefix}%  
  \gls@assign@field{}\@glo@label}{prefixplural}\@glo@entryprefixplural}%
```

If prefixfirst has not been supplied, make it the same as prefix.

```
\expandafter\gls@assign@field\expandafter  
  {\csname glo@\@glo@label @prefix\endcsname}\@glo@label}{prefixfirst}%  
  {\@glo@entryprefixfirst}%
```

If prefixfirstplural has not been supplied, make it the same as prefixplural.

```
\expandafter\gls@assign@field\expandafter  
  {\csname glo@\@glo@label @prefixplural\endcsname}\@glo@label}%  
  {prefixfirstplural}\@glo@entryprefixfirstplural}%  
}
```

Define commands to access these fields:

`\glsentryprefixfirst`

```
\newcommand*\glsentryprefixfirst[1]{\csuse{glo\glsdetoklabel{#1}@prefixfirst}}
```

`\glsentryprefixfirstplural`

```
\newcommand*\glsentryprefixfirstplural[1]{%  
  \csuse{glo\glsdetoklabel{#1}@prefixfirstplural}}
```

`\glsentryprefix`

```
\newcommand*\glsentryprefix[1]{\csuse{glo\glsdetoklabel{#1}@prefix}}
```

`\glsentryprefixplural`

```
\newcommand*\glsentryprefixplural[1]{\csuse{glo\glsdetoklabel{#1}@prefixplural}}
```

Now for the initial upper case variants:

`\Glsentryprefixfirst`

```
\newcommand*\Glsentryprefixfirst[1]{%  
  \@Glsentryfield{#1}{prefixfirst}%  
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentryprefixfirst}{\Glentryprefixfirst}
```

\Glentryprefixfirstplural

```
\newcommand*{\Glentryprefixfirstplural}[1]{%  
  \@Glentryfield{#1}{prefixfirstplural}%  
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentryprefixfirstplural}{\Glentryprefixfirstplural}
```

\Glentryprefix

```
\newcommand*{\Glentryprefix}[1]{%  
  \@Glentryfield{#1}{prefix}%  
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentryprefix}{\Glentryprefix}
```

\Glentryprefixplural

```
\newcommand*{\Glentryprefixplural}[1]{%  
  \@Glentryfield{#1}{prefixplural}%  
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentryprefixplural}{\Glentryprefixplural}
```

Define commands to determine if the prefix keys have been set:

\ifglshasprefix

```
\newcommand*{\ifglshasprefix}[3]{%  
  \ifcempty{glo@glsetoklabel{#1}@prefix}%  
  {#3}%  
  {#2}%  
}
```

\ifglshasprefixplural

```
\newcommand*{\ifglshasprefixplural}[3]{%  
  \ifcempty{glo@glsetoklabel{#1}@prefixplural}%  
  {#3}%  
  {#2}%  
}
```

\ifglshasprefixfirst

```
\newcommand*{\ifglshasprefixfirst}[3]{%  
  \ifcempty{glo@glsetoklabel{#1}@prefixfirst}%  
  {#3}%  
  {#2}%  
}
```

`\ifglshasprefixfirstplural`

```
\newcommand*{\ifglshasprefixfirstplural}[3]{%
  \ifcempty{glo@\glsdetoklabel{#1}@prefixfirstplural}%
  {#3}%
  {#2}%
}
```

`\@glsprefix@record@hook` Need to take into account the possibility that glossaries-extra might be loaded with the record option.

```
\providecommand{\@glsprefix@record@hook}[2]{%
  \ifdef\@glsxtr@record
  {\@glsxtr@record{#1}{#2}{\glslink}}%
  {}%
}
```

`\glsprefixsep` Separator between prefix and term. Does nothing by default.

```
\newcommand{\glsprefixsep}{}
```

Define commands that insert the prefix before commands like `\gls`:

`\pgls`

```
\newrobustcmd{\pgls}{\@gls@hyp@opt\@pgls}
```

`\@pgls` Unstarred version.

```
\newcommand*{\@pgls}[2][{}]{%
  \new@ifnextchar[%
  {\@pgls@{#1}{#2}}%
  {\@pgls@{#1}{#2}[]}%
}
```

`\@pgls@` Read in the final optional argument:

```
\def\@pgls@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglshasprefix{#2}{\glsentryprefix{#2}\glsprefixsep}{}%
    }%
    {\ifglshasprefixfirst{#2}{\glsentryprefixfirst{#2}\glsprefixsep}{}%
    }%
  }%
  \@gls@{#1}{#2}[#3}%
}
```

Similarly for the plural version:

```
\pglsp1
\newrobustcmd{\pglsp1}{\@gls@hyp@opt\@pglsp1}
```

\@pglsp1 Unstarred version.

```
\newcommand*{\@pglsp1}[2][{}]{%
\new@ifnextchar[%
{\@pglsp1@{#1}{#2}}%
{\@pglsp1@{#1}{#2}[]}%
}
```

\@pglsp1@ Read in the final optional argument:

```
\def\@pglsp1@#1#2[#3]{%
\@glsprefix@record@hook{#1}{#2}%
\glsdoifexists{#2}%
{%
\ifglsused{#2}%
{%
\ifgls@hasprefixplural{#2}{\glsentryprefixplural{#2}\glsprefixsep}{}%
}%
{%
\ifgls@hasprefixfirstplural{#2}%
{\glsentryprefixfirstplural{#2}\glsprefixsep}{}%
}%
\@glspl@{#1}{#2}[#3]%
}%
}
```

Now for the first letter upper case versions:

```
\PglS
\newrobustcmd{\PglS}{\@gls@hyp@opt\@PglS}
```

Add sentence case mapping.

```
\glsmfuaddmap{\pgls}{\PglS}
```

\@PglS Unstarred version.

```
\newcommand*{\@PglS}[2][{}]{%
\new@ifnextchar[%
{\@PglS@{#1}{#2}}%
{\@PglS@{#1}{#2}[]}%
}
```

\@PglS@ Read in the final optional argument:

```
\def\@PglS@#1#2[#3]{%
\@glsprefix@record@hook{#1}{#2}%
\glsdoifexists{#2}%
{%
\ifglsused{#2}%
}
```

```

{%
  \ifglshasprefix{#2}%
  {%
    \Glentryprefix{#2}%
    \glsprefixsep
    \@gls@{#1}{#2}[#3]%
  }%
  {\@Gls@{#1}{#2}[#3]}%
}%
{%
  \ifglshasprefixfirst{#2}%
  {%
    \Glentryprefixfirst{#2}%
    \glsprefixsep
    \@gls@{#1}{#2}[#3]%
  }%
  {\@Gls@{#1}{#2}[#3]}%
}%
}%
}

```

Similarly for the plural version:

```

\Pglsp1
\newrobustcmd{\Pglsp1}{\@gls@hyp@opt\Pglsp1}

```

Add sentence case mapping.

```

\glsmfuaddmap{\pglsp1}{\Pglsp1}

```

\@Pglsp1 Unstarred version.

```

\newcommand*{\@Pglsp1}[2][{}]{%
  \new@ifnextchar[%
    {\@Pglsp1@{#1}{#2}}%
    {\@Pglsp1@{#1}{#2}[{}]}%
}

```

\@Pglsp1@ Read in the final optional argument:

```

\def\@Pglsp1@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglshasprefixplural{#2}%
      {%
        \Glentryprefixplural{#2}%
        \glsprefixsep
        \@glspl@{#1}{#2}[#3]%
      }%
    }%
  }%
}

```

```

        {\@Glspl@{#1}{#2}[#3]}%
    }%
    {%
        \ifglshasprefixfirstplural{#2}%
        {%
            \Glsentryprefixfirstplural{#2}%
            \glsprefixsep
            \@glspl@{#1}{#2}[#3]%
        }%
        {\@Glspl@{#1}{#2}[#3]}%
    }%
}
}

```

Finally the all upper case versions:

```

\PGLS
\newrobustcmd{\PGLS}{\@gls@hyp@opt\@PGLS}

```

\@PGLS Unstarred version.

```

\newcommand*{\@PGLS}[2] [] {%
    \new@ifnextchar [%
        {\@PGLS@{#1}{#2}}%
        {\@PGLS@{#1}{#2} []}%
    }
}

```

\@PGLS@ Read in the final optional argument:

```

\def\@PGLS@#1#2[#3]{%
    \@glsprefix@record@hook{#1}{#2}%
    \glsdoifexists{#2}%
    {%
        \ifglsused{#2}%
        {%
            \ifglshasprefix{#2}%
            {\glsuppercase{\glsentryprefix{#2}\glsprefixsep}}{%
            }%
        }%
        {%
            \ifglshasprefixfirst{#2}%
            {\glsuppercase{\glsentryprefixfirst{#2}\glsprefixsep}}{%
            }%
        }%
        \@GLS@{#1}{#2}[#3]%
    }%
}
}

```

Plural version:

```

\PGLSpl
\newrobustcmd{\PGLSpl}{\@gls@hyp@opt\@PGLSpl}

```


\@PGLSp1 Unstarred version.

```
\newcommand*{\@PGLSp1}[2][ ]{%
  \new@ifnextchar[%
    {\@PGLSp1@{#1}{#2}}%
    {\@PGLSp1@{#1}{#2}[ ]}%
}
```

\@PGLSp1@ Read in the final optional argument:

```
\def\@PGLSp1@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \@glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglschasprefixplural{#2}%
        {\glsuppercase{\glsentryprefixplural{#2}\glsprefixsep}}{%
      }%
    }%
    \ifglschasprefixfirstplural{#2}%
      {\glsuppercase{\glsentryprefixfirstplural{#2}\glsprefixsep}}{%
    }%
  }%
  \@GLSp1@{#1}{#2}[#3]%
}%
}
```

6 Accessibility Support (glossaries-accsupp Code)

The glossaries-accsupp package is experimental. It is intended to provide a means of using the PDF accessibility support in glossary entries. See the accsupp documentation for further details about accessibility support.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossaries-accsupp-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-accsupp-2021-11-01.sty}
\DeclareCurrentRelease{v4.56}{2025-03-03}
```

Package version number now in line with main glossaries package number.

```
\ProvidesPackage{glossaries-accsupp}[2025/03/03 v4.56 (NLCT)
  Experimental glossaries accessibility]
```

Pass all options to glossaries:

```
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
```

Process options:

```
\ProcessOptions
```

This package should be loaded before glossaries-extra, so complain if that has already been loaded.

```
\@ifpackageloaded{glossaries-extra}
{%
```

If the `accsupp` option was used, `\@glsxtr@doaccsupp` will have been set, otherwise it will be empty.

```
\ifx\@glsxtr@doaccsupp\empty
  \GlossariesWarning{The ‘glossaries-accsupp’
package has been loaded\MessageBreak
after the ‘glossaries-extra’ package. This\MessageBreak
can cause a failure to integrate both packages. \MessageBreak
Either use the ‘accsupp’ option when you load\MessageBreak
‘glossaries-extra’ or load ‘glossaries-accsupp’\MessageBreak
before loading ‘glossaries-extra’}%
\fi
}
{}
```

`\compatibleglossentry` Override style compatibility macros:

`\compatiblesubglossentry`

Required packages:

```
\RequirePackage{glossaries}
```

`\gls@accsupp@engine` There’s currently only support for `accsupp`, but if you define `\gls@accsupp@engine` before loading `glossaries-accsupp`, you can prevent `accsupp` from being loaded. Redefining this command after `glossaries-accsupp` has loaded obviously won’t do anything (and so is an internal command to deter casual use). If it is defined to something other than `accsupp` then `\gls@accessibility` will need to be defined to something appropriate.

```
\providecommand{\gls@accsupp@engine}{accsupp}
```

```
\gls@accessibility[<options>]{<PDF element>}{<value>}
{<content>}
```

`\gls@accessibility`

```
\providecommand{\gls@accessibility}[4]{#4}
\ifdefstring\gls@accsupp@engine{accsupp}
{
  \RequirePackage{accsupp}
  \renewcommand{\gls@accessibility}[4]{%
    \BeginAccSupp{#1,#2={#3}}#4\EndAccSupp{}}%
  }
}
{}
```

```
\glsaccessibility[<options>]{<PDF element>}{<value>}
{<content>}
```

`\glsaccessibility`

User-level command that includes debug info if required.

```
\newcommand{\glsaccessibility}[4] []{%  
  \glsshowaccsupp{#1}{#2}{#3}%  
  \gls@accessibility{#1}{#2}{#3}{#4}%  
}
```

6.1 Defining Replacement Text

The version 0.1 stored the replacement text in the symbol key. This has been changed to use the new keys defined here. Example of use:

```
%\newglossaryentry{dr}{name=Dr,description={},access={Doctor}}  
%
```

access The replacement text corresponding to the name key:

```
\define@key{glossentry}{access}{%  
  \def\@glo@access{#1}%  
}
```

textaccess The replacement text corresponding to the text key:

```
\define@key{glossentry}{textaccess}{%  
  \def\@glo@textaccess{#1}%  
}
```

firstaccess The replacement text corresponding to the first key:

```
\define@key{glossentry}{firstaccess}{%  
  \def\@glo@firstaccess{#1}%  
}
```

pluralaccess The replacement text corresponding to the plural key:

```
\define@key{glossentry}{pluralaccess}{%  
  \def\@glo@pluralaccess{#1}%  
}
```

firstpluralaccess The replacement text corresponding to the firstplural key:

```
\define@key{glossentry}{firstpluralaccess}{%  
  \def\@glo@firstpluralaccess{#1}%  
}
```

symbolaccess The replacement text corresponding to the symbol key:

```
\define@key{glossentry}{symbolaccess}{%  
  \def\@glo@symbolaccess{#1}%  
}
```

symbolpluralaccess The replacement text corresponding to the symbolplural key:

```
\define@key{glossentry}{symbolpluralaccess}{%  
  \def\@glo@symbolpluralaccess{#1}%  
}
```

descriptionaccess The replacement text corresponding to the description key:

```
\define@key{glossentry}{descriptionaccess}{%  
  \def\@glo@descaccess{#1}%  
}
```

descriptionpluralaccess The replacement text corresponding to the descriptionplural key:

```
\define@key{glossentry}{descriptionpluralaccess}{%  
  \def\@glo@descpluralaccess{#1}%  
}
```

shortaccess The replacement text corresponding to the short key:

```
\define@key{glossentry}{shortaccess}{%  
  \def\@glo@shortaccess{#1}%  
}
```

shortpluralaccess The replacement text corresponding to the shortplural key:

```
\define@key{glossentry}{shortpluralaccess}{%  
  \def\@glo@shortpluralaccess{#1}%  
}
```

longaccess The replacement text corresponding to the long key:

```
\define@key{glossentry}{longaccess}{%  
  \def\@glo@longaccess{#1}%  
}
```

longpluralaccess The replacement text corresponding to the longplural key:

```
\define@key{glossentry}{longpluralaccess}{%  
  \def\@glo@longpluralaccess{#1}%  
}
```

There are now also keys that correspond to the user keys:

user1access The replacement text corresponding to the user1 key:

```
\define@key{glossentry}{user1access}{%  
  \def\@glo@useriaccess{#1}%  
}
```

user2access The replacement text corresponding to the user2 key:

```
\define@key{glossentry}{user2access}{%  
  \def\@glo@useriiaccess{#1}%  
}
```

user3access The replacement text corresponding to the user3 key:

```
\define@key{glossentry}{user3access}{%  
  \def\@glo@useriiiaccess{#1}%  
}
```

`user4access` The replacement text corresponding to the `user4` key:

```
\define@key{glossentry}{user4access}{%
  \def\@glo@userivaccess{#1}%
}
```

`user5access` The replacement text corresponding to the `user5` key:

```
\define@key{glossentry}{user5access}{%
  \def\@glo@uservaccess{#1}%
}
```

`user6access` The replacement text corresponding to the `user6` key:

```
\define@key{glossentry}{user6access}{%
  \def\@glo@userviaccess{#1}%
}
```

For any custom keys, the replacement text would have to be explicitly put in the value, e.g., `user1={\glsshortaccsupp{inches}{in}}`.

Append these new keys to `\@gls@keymap`:

```
\appto\@gls@keymap{,%
  {access}{access},%
  {textaccess}{textaccess},%
  {firstaccess}{firstaccess},%
  {pluralaccess}{pluralaccess},%
  {firstpluralaccess}{firstpluralaccess},%
  {symbolaccess}{symbolaccess},%
  {symbolpluralaccess}{symbolpluralaccess},%

  {descriptionaccess}{descaccess},%
  {descriptionpluralaccess}{descpluralaccess},%
  {shortaccess}{shortaccess},%
  {shortpluralaccess}{shortpluralaccess},%
  {longaccess}{longaccess},%
  {longpluralaccess}{longpluralaccess},%
  {user1access}{useriaccess},%
  {user2access}{useriiaccess},%
  {user3access}{useriiiaccess},%
  {user4access}{userivaccess},%
  {user5access}{uservaccess},%
  {user6access}{userviaccess}%
}
```

`\@gls@noaccess` Indicates that no replacement text has been provided.

```
\def\@gls@noaccess{\relax}
```

Previously, the access key was initialised to the value of the `symbol` key at the start for backwards compatibility. This causes a problem for situations where the replacement text is provided for `symbol` but not for `name` so this behaviour has been removed.

```
\let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
```

```

\renewcommand*{\@newglossaryentryprehook}{%
  \@gls@oldnewglossaryentryprehook
  \def\@glo@access{\relax}%

```

Initialise the other keys:

```

\def\@glo@textaccess{\@glo@access}%
\def\@glo@firstaccess{\@glo@access}%
\def\@glo@pluralaccess{\@glo@textaccess}%
\def\@glo@firstpluralaccess{\@glo@pluralaccess}%
\def\@glo@symbolaccess{\relax}%
\def\@glo@symbolpluralaccess{\@glo@symbolaccess}%
\def\@glo@descaccess{\relax}%
\def\@glo@descpluralaccess{\@glo@descaccess}%
\def\@glo@shortaccess{\relax}%
\def\@glo@shortpluralaccess{\@glo@shortaccess}%
\def\@glo@longaccess{\relax}%
\def\@glo@longpluralaccess{\@glo@longaccess}%
\def\@glo@useriaccess{\relax}%
\def\@glo@useriiaaccess{\relax}%
\def\@glo@useriiaaaccess{\relax}%
\def\@glo@userivaccess{\relax}%
\def\@glo@uservaccess{\relax}%
\def\@glo@userviaccess{\relax}%
}

```

Add to the end hook:

```

\let\@gls@oldnewglossaryentryposthook\@newglossaryentryposthook
\renewcommand*{\@newglossaryentryposthook}{%
  \@gls@oldnewglossaryentryposthook

```

Store the access information:

```

\expandafter
  \protected@xdef\csname glo@\@glo@label @access\endcsname{%
    \@glo@access}%
\expandafter
  \protected@xdef\csname glo@\@glo@label @textaccess\endcsname{%
    \@glo@textaccess}%
\expandafter
  \protected@xdef\csname glo@\@glo@label @firstaccess\endcsname{%
    \@glo@firstaccess}%
\expandafter
  \protected@xdef\csname glo@\@glo@label @pluralaccess\endcsname{%
    \@glo@pluralaccess}%
\expandafter
  \protected@xdef\csname glo@\@glo@label @firstpluralaccess\endcsname{%
    \@glo@firstpluralaccess}%
\expandafter
  \protected@xdef\csname glo@\@glo@label @symbolaccess\endcsname{%
    \@glo@symbolaccess}%
\expandafter
  \protected@xdef\csname glo@\@glo@label @symbolpluralaccess\endcsname{%

```

```

        \@glo@symbolpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descaccess\endcsname{%
    \@glo@descaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descpluralaccess\endcsname{%
    \@glo@descpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortaccess\endcsname{%
    \@glo@shortaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortpluralaccess\endcsname{%
    \@glo@shortpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longaccess\endcsname{%
    \@glo@longaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longpluralaccess\endcsname{%
    \@glo@longpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriaccess\endcsname{%
    \@glo@useriaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiaccess\endcsname{%
    \@glo@useriiaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiiaccess\endcsname{%
    \@glo@useriiiaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @userivaccess\endcsname{%
    \@glo@userivaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @uservaccess\endcsname{%
    \@glo@uservaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @userviaccess\endcsname{%
    \@glo@userviaccess}%
}

```

6.2 Accessing Replacement Text

`\glsentryaccess` Get the value of the access key for the entry with the given label:

```

\newcommand*{\glsentryaccess}[1]{%
    \@gls@entry@field{#1}{access}%
}

```

`\glsentrytextaccess` Get the value of the textaccess key for the entry with the given label:

```

\newcommand*{\glsentrytextaccess}[1]{%
    \@gls@entry@field{#1}{textaccess}%
}

```

}

`\glsentryfirstaccess` Get the value of the `firstaccess` key for the entry with the given label:

```
\newcommand*{\glsentryfirstaccess}[1]{%
  \@gls@entry@field{#1}{firstaccess}%
}
```

`\glsentrypluralaccess` Get the value of the `pluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentrypluralaccess}[1]{%
  \@gls@entry@field{#1}{pluralaccess}%
}
```

`\glsentryfirstpluralaccess` Get the value of the `firstpluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentryfirstpluralaccess}[1]{%
  \@gls@entry@field{#1}{firstpluralaccess}%
}
```

`\glsentrysymbolaccess` Get the value of the `symbolaccess` key for the entry with the given label:

```
\newcommand*{\glsentrysymbolaccess}[1]{%
  \@gls@entry@field{#1}{symbolaccess}%
}
```

`\glsentrysymbolpluralaccess` Get the value of the `symbolpluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentrysymbolpluralaccess}[1]{%
  \@gls@entry@field{#1}{symbolpluralaccess}%
}
```

`\glsentrydescaccess` Get the value of the `descriptionaccess` key for the entry with the given label:

```
\newcommand*{\glsentrydescaccess}[1]{%
  \@gls@entry@field{#1}{descaccess}%
}
```

`\glsentrydescpluralaccess` Get the value of the `descriptionpluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentrydescpluralaccess}[1]{%
  \@gls@entry@field{#1}{descpluralaccess}%
}
```

`\glsentryshortaccess` Get the value of the `shortaccess` key for the entry with the given label:

```
\newcommand*{\glsentryshortaccess}[1]{%
  \@gls@entry@field{#1}{shortaccess}%
}
```

`\glsentryshortpluralaccess` Get the value of the `shortpluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentryshortpluralaccess}[1]{%
  \@gls@entry@field{#1}{shortpluralaccess}%
}
```


`\glentrylongaccess` Get the value of the `longaccess` key for the entry with the given label:

```

\newcommand*{\glentrylongaccess}[1]{%
  \@gls@entry@field{#1}{longaccess}%
}

```

`\glentrylongpluralaccess` Get the value of the `longpluralaccess` key for the entry with the given label:

```

\newcommand*{\glentrylongpluralaccess}[1]{%
  \@gls@entry@field{#1}{longpluralaccess}%
}

```

`\glentryuseriaccess` Get the value of the `user1access` key for the entry with the given label:

```

\newcommand*{\glentryuseriaccess}[1]{%
  \@gls@entry@field{#1}{useriaccess}%
}

```

`\glentryuseriiaccess` Get the value of the `user2access` key for the entry with the given label:

```

\newcommand*{\glentryuseriiaccess}[1]{%
  \@gls@entry@field{#1}{useriiaccess}%
}

```

`\glentryuseriiiaccess` Get the value of the `user3access` key for the entry with the given label:

```

\newcommand*{\glentryuseriiiaccess}[1]{%
  \@gls@entry@field{#1}{useriiiaccess}%
}

```

`\glentryuserivaccess` Get the value of the `user4access` key for the entry with the given label:

```

\newcommand*{\glentryuserivaccess}[1]{%
  \@gls@entry@field{#1}{userivaccess}%
}

```

`\glentryuservaccess` Get the value of the `user5access` key for the entry with the given label:

```

\newcommand*{\glentryuservaccess}[1]{%
  \@gls@entry@field{#1}{uservaccess}%
}

```

`\glentryuserviaccess` Get the value of the `user6access` key for the entry with the given label:

```

\newcommand*{\glentryuserviaccess}[1]{%
  \@gls@entry@field{#1}{userviaccess}%
}

```

There are three types of replacement text:

Alt Description of some content that's non-textual (for example, an image). A word break is assumed after the content.

ActualText A character or sequence of characters that replaces textual content (for example, a dropped capital, a ligature or a symbol). No word break is assumed after the content.

E Expansion of an abbreviation to avoid ambiguity (for example, “St” could be short for “saint” or “street”).

Therefore, rather than having one command for all fields, it’s better to have a command dependent on the field type. For example, the `short` and `shortpl` keys would require `E`, the `symbol` key would require `ActualText`, and a field that contains an image would require `Alt`.

```
\glsfieldaccsupp{<replacement>}{<content>}{<field>}{<label>}
```

`\glsfieldaccsupp`

Test if there’s a command called `\gls<field>accsupp`. If there is then use that otherwise use `\glsaccsupp`. The first argument should be the internal field label (not the key). The final argument is the entry label. If `glossaries-extra` has been loaded, this first checks for `\glsxtr<category><field>accsupp` and `\glsxtr<category>accsupp`.

```
\newcommand{\glsfieldaccsupp}[4]{%
  \ifdef\glscategory
  {%
    \ifcsdef{glsxtr\glscategory{#4}#3accsupp}%
    {\csname glsxtr\glscategory{#4}#3accsupp\endcsname{#1}{#2}}%
    {%
      \ifcsdef{glsxtr\glscategory{#4}accsupp}%
      {\csname glsxtr\glscategory{#4}accsupp\endcsname{#1}{#2}}%
      {%
        \ifcsdef{gls#3accsupp}%
        {\csname gls#3accsupp\endcsname{#1}{#2}}%
        {\glsaccsupp{#1}{#2}}%
      }%
    }%
  }%
  {%
    \ifcsdef{gls#3accsupp}%
    {\csname gls#3accsupp\endcsname{#1}{#2}}%
    {\glsaccsupp{#1}{#2}}%
  }%
}
```

```
\xglsfieldaccsupp{<replacement>}{<content>}{<field>}{<label>}
```

`\xglsfieldaccsupp`

As `\glsfieldaccsupp` but fully expand replacement text.

```
\newcommand{\xglsfieldaccsupp}[1]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsfieldaccsupp\expandafter{\@gls@replacementtext}%
}
```

`\glsshortaccsupp`

```
\glsshortaccsupp{<replacement text>}{<text>}
```

```
\newcommand*{\glsshortaccsupp}[2]{\glsaccessibility{E}{#1}{#2}}
```

`\glsshortplaccsupp`

```
\glsshortplaccsupp{<replacement text>}{<text>}
```

```
\newcommand*{\glsshortplaccsupp}{\glsshortaccsupp}
```

`\glsaccsupp`

```
\glsaccsupp{<replacement text>}{<text>}
```

```
\newcommand*{\glsaccsupp}[2]{\glsaccessibility{ActualText}{#1}{#2}}
```

`\xglsaccsupp` Fully expands replacement text before calling `\glsaccsupp`

```
\newcommand*{\xglsaccsupp}[2]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsaccsupp\expandafter{\@gls@replacementtext}{#2}%
}
```

`\@gls@access@display` Deprecated in v4.45. Use `\@gls@fieldaccess@display` instead.

```
\newcommand*{\@gls@access@display}[2]{%
  \protected@edef\@glo@access{#2}%
  \ifx\@glo@access\@gls@noaccess
    #1%
  \else
    \xglsaccsupp{\@glo@access}{#1}%
  \fi
}
```

`\@gls@fieldaccess@display`

```
\@gls@fieldaccess@display{<label>}{<field>}{<content>}
{<replacement>}
```

```
\newcommand*{\@gls@fieldaccess@display}[4]{%
  \protected@edef\@glo@access{#4}%
  \ifdefequal\@glo@access\@gls@noaccess
    {#3}%
  {\expandafter\glsfieldaccsupp\expandafter{\@glo@access}{#3}{#2}{#1}}%
}
```

`\glsnameaccessdisplay` Displays the first argument with the accessibility text for the entry with the label given by the second argument (if set).

```
\newrobustcmd*{\glsnameaccessdisplay}[2]{%
  \ifcsundef{glo@glsetoklabel{#2}@access}%
```

```

        {#1}%
        {\@gls@fieldaccess@display{#2}{name}{#1}{\glsentryaccess{#2}}}%
    }

\glstextaccessdisplay As above but for the textaccess replacement text.
\newrobustcmd*{\glstextaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@textaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{text}{#1}{\glsentrytextaccess{#2}}}%
}

\glspluralaccessdisplay As above but for the pluralaccess replacement text.
\newrobustcmd*{\glspluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@pluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{plural}{#1}{\glsentrypluralaccess{#2}}}%
}

\glsfirstaccessdisplay As above but for the firstaccess replacement text.
\newrobustcmd*{\glsfirstaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@firstaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{first}{#1}{\glsentryfirstaccess{#2}}}%
}

\glsfirstpluralaccessdisplay As above but for the firstpluralaccess replacement text.
\newrobustcmd*{\glsfirstpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@firstpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{firstpl}{#1}{\glsentryfirstpluralaccess{#2}}}%
}

\glssymbolaccessdisplay As above but for the symbolaccess replacement text.
\newrobustcmd*{\glssymbolaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@symbolaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{symbol}{#1}{\glsentrysymbolaccess{#2}}}%
}

\glssymbolpluralaccessdisplay As above but for the symbolpluralaccess replacement text.
\newrobustcmd*{\glssymbolpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@symbolpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{symbolplural}{#1}{\glsentrysymbolpluralaccess{#2}}}%
}

\glsdescriptionaccessdisplay As above but for the descriptionaccess replacement text.
\newrobustcmd*{\glsdescriptionaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@descaccess}%

```

```

        {#1}%
        {\@gls@fieldaccess@display{#2}{desc}{#1}{\glsentrydescaccess{#2}}}%
    }
descriptionpluralaccessdisplay As above but for the descriptionpluralaccess replacement text.
\newrobustcmd*{\glsdescriptionpluralaccessdisplay}[2]{%
\ifcsundef{glo@\glsdetoklabel{#2}@descpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{descplural}{#1}{\glsentrydescpluralaccess{#2}}}%
}
\glsshortaccessdisplay As above but for the shortaccess replacement text.
\newrobustcmd*{\glsshortaccessdisplay}[2]{%
\ifcsundef{glo@\glsdetoklabel{#2}@shortaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{short}{#1}{\glsentryshortaccess{#2}}}%
}
\glsshortpluralaccessdisplay As above but for the shortpluralaccess replacement text.
\newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
\ifcsundef{glo@\glsdetoklabel{#2}@shortpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glsentryshortpluralaccess{#2}}}%
}
\glslongaccessdisplay As above but for the longaccess replacement text.
\newrobustcmd*{\glslongaccessdisplay}[2]{%
\ifcsundef{glo@\glsdetoklabel{#2}@longaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{long}{#1}{\glsentrylongaccess{#2}}}%
}
\glslongpluralaccessdisplay As above but for the longpluralaccess replacement text.
\newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
\ifcsundef{glo@\glsdetoklabel{#2}@longpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{longpl}{#1}{\glsentrylongpluralaccess{#2}}}%
}
\glsuseriaccessdisplay As above but for the user1access replacement text.
\newrobustcmd*{\glsuseriaccessdisplay}[2]{%
\ifcsundef{glo@\glsdetoklabel{#2}@useriaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useri}{#1}{\glsentryuseriaccess{#2}}}%
}
\glsuseriiaccessdisplay As above but for the user2access replacement text.
\newrobustcmd*{\glsuseriiaccessdisplay}[2]{%
\ifcsundef{glo@\glsdetoklabel{#2}@useriiaccess}%

```

```

    {#1}%
    {\@gls@fieldaccess@display{#2}{userii}{#1}{\glsentryuseriiaccess{#2}}}%
  }

```

`\glsuseriiiaccessdisplay` As above but for the user3access replacement text.

```

\newrobustcmd*{\glsuseriiiaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#2}@useriiiaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useriii}{#1}{\glsentryuseriiiaccess{#2}}}%
}

```

`\glsuserivaccessdisplay` As above but for the user4access replacement text.

```

\newrobustcmd*{\glsuserivaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#2}@userivaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useriv}{#1}{\glsentryuserivaccess{#2}}}%
}

```

`\glsuservaccessdisplay` As above but for the user5access replacement text.

```

\newrobustcmd*{\glsuservaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#2}@uservaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{userv}{#1}{\glsentryuservaccess{#2}}}%
}

```

`\glsuserviaccessdisplay` As above but for the user6access replacement text.

```

\newrobustcmd*{\glsuserviaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#2}@userviaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{uservi}{#1}{\glsentryuserviaccess{#2}}}%
}

```

`\glsaccessdisplay` Gets the replacement text corresponding to the named key given by the first argument and calls the appropriate command defined above.

```

\newrobustcmd*{\glsaccessdisplay}[3]{%
  \ifcsundef{gls#1accessdisplay}%
  {%
    \PackageError{glossaries-accsupp}{No accessibility support
      for key ‘#1’}{%
    }%
  }%
  \csname gls#1accessdisplay\endcsname{#2}{#3}%
}

```

`\@gls@default@entryfmt` Deprecated `\@gls@default@entryfmt` was redefined to use accessibility information but has been removed in v4.50.

```

\glsentryfmt Redefine to use accessibility information.
\renewcommand*\glsentryfmt{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      Plural form
      \glscapscase
      {%
        Don't adjust case
        \ifglsused\glslabel
        {%
          Subsequent use
          \glspluralaccessdisplay
            {\glsentryplural{\glslabel}}{\glslabel}%
          \glsinsert
          }%
          {%
            First use
            \glsfirstpluralaccessdisplay
              {\glsentryfirstplural{\glslabel}}{\glslabel}%
            \glsinsert
            }%
            }%
            {%
              Make first letter upper case
              \ifglsused\glslabel
              {%
                Subsequent use.
                \glspluralaccessdisplay
                  {\Glsentryplural{\glslabel}}{\glslabel}%
                \glsinsert
                }%
                {%
                  First use
                  \glsfirstpluralaccessdisplay
                    {\Glsentryfirstplural{\glslabel}}{\glslabel}%
                  \glsinsert
                  }%
                  }%
                  {%
                    Make all upper case
                    \ifglsused\glslabel
                    {%

```

Subsequent use

```
\glsppluralaccessdisplay
  {\glssupercase{\glstentryplural{\glslabel}}}%
  {\glslabel}%
\glssupercase{\glsinsert}%
}%
{%
```

First use

```
\glsfirstpluralaccessdisplay
  {\glssupercase{\glstentryfirstplural{\glslabel}}}%
  {\glslabel}%
\glssupercase{\glsinsert}%
}%
}%
}%
{%
```

Singular form

```
\glscapscale
{%
```

Don't adjust case

```
\ifglsused\glslabel
{%
```

Subsequent use

```
\glstextaccessdisplay{\glstentrytext{\glslabel}}{\glslabel}%
\glsinsert
}%
{%
```

First use

```
\glsfirstaccessdisplay{\glstentryfirst{\glslabel}}{\glslabel}%
\glsinsert
}%
}%
{%
```

Make first letter upper case

```
\ifglsused\glslabel
{%
```

Subsequent use

```
\glstextaccessdisplay{\Glstentrytext{\glslabel}}{\glslabel}%
\glsinsert
}%
{%
```

First use

```
\glsfirstaccessdisplay{\Glstentryfirst{\glslabel}}{\glslabel}%
\glsinsert
}%
```



```
}%  
{%
```

Make all upper case

```
\ifglsused\glslabel  
{%
```

Subsequent use

```
\glstextaccessdisplay  
  {\glsuppercase{\glsentrytext{\glslabel}}}{\glslabel}%  
\glsuppercase{\glsinsert}%  
}%  
{%
```

First use

```
\glsfirstaccessdisplay  
  {\glsuppercase{\glsentryfirst{\glslabel}}}{\glslabel}%  
\glsuppercase{\glsinsert}%  
}%  
}%  
}%  
{%
```

Custom text provided in `\glsdisp`. (The insert should be empty at this point.)
The accessibility information, if required, will have to be explicitly included in
the custom text.

```
\glscustomtext\glsinsert  
}%  
}
```

`\glsngenacfmt` Redefine to include accessibility information.

```
\renewcommand*{\glsngenacfmt}{%  
  \ifdefempty\glscustomtext  
  {%  
    \ifglsused\glslabel  
    {%
```

Subsequent use:

```
\glsifplural  
{%
```

Subsequent plural form:

```
\glsupcase  
{%
```

Subsequent plural form, don't adjust case:

```
\acronymfont  
  {\glsshortpluralaccessdisplay  
   {\glsentryshortpl{\glslabel}}{\glslabel}}%  
\glsinsert  
}%  
{%
```

Subsequent plural form, make first letter upper case:

```
\acronymfont
  {\glsshortpluralaccessdisplay
   {\Glsentryshortpl{\glslabel}}{\glslabel}}%
\glsinsert
}%
{%
```

Subsequent plural form, all caps:

```
\glsuppercase
  {\acronymfont
   {\glsshortpluralaccessdisplay
    {\Glsentryshortpl{\glslabel}}{\glslabel}}%
   \glsinsert}%
}%
{%
```

Subsequent singular form

```
\glscapscase
{%
```

Subsequent singular form, don't adjust case:

```
\acronymfont
  {\glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
\glsinsert
}%
{%
```

Subsequent singular form, make first letter upper case:

```
\acronymfont
  {\glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
\glsinsert
}%
{%
```

Subsequent singular form, all caps:

```
\glsuppercase
  {\acronymfont{%
   \glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
   \glsinsert}%
}%
}%
{%
```

First use:

```
\glsifplural
{%
```

First use plural form:

```
\glscapscase
{%
```

First use plural form, don't adjust case:

```
\genplacrfullformat{\glslabel}{\glsinsert}%  
}%  
{%
```

First use plural form, make first letter upper case:

```
\Genplacrfullformat{\glslabel}{\glsinsert}%  
}%  
{%
```

First use plural form, all caps:

```
\glsuppercase  
{\genplacrfullformat{\glslabel}{\glsinsert}}%  
}%  
}%  
{%
```

First use singular form

```
\glsescapscase  
{%
```

First use singular form, don't adjust case:

```
\genacrfullformat{\glslabel}{\glsinsert}%  
}%  
{%
```

First use singular form, make first letter upper case:

```
\Genacrfullformat{\glslabel}{\glsinsert}%  
}%  
{%
```

First use singular form, all caps:

```
\glsuppercase  
{\genacrfullformat{\glslabel}{\glsinsert}}%  
}%  
}%  
}%  
{%
```

User supplied text. (The insert should be empty at this point.) The accessibility information, if required, will have to be explicitly included in the custom text.

```
\glscustomtext  
}%  
}
```

`\genacrfullformat` Redefine to include accessibility information.

```
\renewcommand*{\genacrfullformat}[2]{%  
  \glslongaccessdisplay{\glsentrylong{#1}}{#1}#2\space  
  (\glsshortaccessdisplay{\protect\firstacronymfont{\glsentryshort{#1}}}{#1})%  
}
```

```

\Genacrfullformat Redefine to include accessibility information.
  \renewcommand*{\Genacrfullformat}[2]{%
    \glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space
    (\glsshortaccessdisplay{\protect\firstacronymfont{\Glsentryshort{#1}}}{#1})%
  }

\genplacrfullformat Redefine to include accessibility information.
  \renewcommand*{\genplacrfullformat}[2]{%
    \glslongpluralaccessdisplay{\glsentrylongpl{#1}}{#1}#2\space
    (\glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
  }

\Genplacrfullformat Redefine to include accessibility information.
  \renewcommand*{\Genplacrfullformat}[2]{%
    \glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}#2\space
    (\glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
  }

\@acrshort
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\glsentryshort{#2}}{#2}}#3%
    }%
  }

Call \@gls@link
  \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%

  \glspostlinkhook
}

\@Acrshort
\def\@Acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glsapspace\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
    }%
  }

```

```

Call \@gls@link
  \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}

\@ACRshort
\def\@ACRshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax

    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay
        {\glsuppercase{\glsentryshort{#2}}}{#2}}#3%
    }%
  }%
Call \@gls@link
  \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}

\@acrlong
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax

    \let\glsifplural\@secondoftwo
    \let\glsapsase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{\glsentrylong{#2}}{#2}}#3%
    }%
  }%
Call \@gls@link
  \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}

\@Acrlong
\def\@Acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax

```



```

\@glsfirst@
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\glsentryfirst{#2}}{#2}#3}%
}

\@Glsfirst@
\def\@Glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\Glsentryfirst{#2}}{#2}#3}%
}

\@GLSfirst@
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsfirstaccessdisplay{\glsupercase{\glsentryfirst{#2}}}{#2}%
  \glsupercase{#3}}%
}

\@glsplural@
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\glsentryplural{#2}}{#2}#3}%
}

\@Glsplural@
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\Glsentryplural{#2}}{#2}#3}%
}

\@GLSplural@
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glspluralaccessdisplay{\glsupercase{\glsentryplural{#2}}}{#2}%
  \glsupercase{#3}}%
}

\@glsfirstplural@
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\glsentryfirstplural{#2}}{#2}#3}%
}

\@Glsfirstplural@
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\Glsentryfirstplural{#2}}{#2}#3}%
}

\@GLSfirstplural@
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsfirstpluralaccessdisplay{\glsupercase{\glsentryfirstplural{#2}}}{#2}%
  \glsupercase{#3}}%
}

```

```

\@glsname@
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glsentryname{#2}}{#2}#3}%
}

\@Glsname@
\def\@Glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\Glsentryname{#2}}{#2}#3}%
}

\@GLSname@
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsnameaccessdisplay{\glsuppercase{\glsentryname{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

\@glsdesc@
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\glsentrydesc{#2}}{#2}#3}%
}

\@Glsdesc@
\def\@Glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
}

\@GLSdesc@
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsdescriptionaccessdisplay{\glsuppercase{\glsentrydesc{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

\@glsdescplural@
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\glsentrydescplural{#2}}{#2}#3}%
}

\@Glsdescplural@
\def\@Glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
}

\@GLSdescplural@
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsdescriptionpluralaccessdisplay{\glsuppercase{\glsentrydescplural{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

```



```

\@glssymbol@
\def\@glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
}

\@Glsymbol@
\def\@Glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}

\@GLSsymbol@
\def\@GLSsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glssymbolaccessdisplay{\glsupercase{\glsentrysymbol{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsymbolplural@
\def\@glsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsymbolpluralaccessdisplay{\glsentrysymbolplural{#2}}{#2}#3}%
}

\@Glsymbolplural@
\def\@Glsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsymbolpluralaccessdisplay{\Glsentrysymbolplural{#2}}{#2}#3}%
}

\@GLSsymbolplural@
\def\@GLSsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsymbolpluralaccessdisplay{\glsupercase{\glsentrysymbolplural{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsuseri@
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glsentryuseri{#2}}{#2}#3}%
}

\@Glsuseri@
\def\@Glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
}

\@GLSuseri@
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriaccessdisplay{\glsupercase{\glsentryuseri{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

```

```

\@glsuserii@
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glsentryuserii{#2}}{#2}#3}%
}

\@Glsuserii@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glsentryuserii{#2}}{#2}#3}%
}

\@GLSuserii@
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiaccessdisplay{\glsupercase{\glsentryuserii{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsuseriii@
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\glsentryuseriii{#2}}{#2}#3}%
}

\@Glsuseriii@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glsentryuseriii{#2}}{#2}#3}%
}

\@GLSuseriii@
\def\@GLSuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiiaccessdisplay{\glsupercase{\glsentryuseriii{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsuseriv@
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glsentryuseriv{#2}}{#2}#3}%
}

\@Glsuseriv@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
}

\@GLSuseriv@
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuserivaccessdisplay{\glsupercase{\glsentryuseriv{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

```

```

\@glsuserv@
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\glsentryuserv{#2}}{#2}#3}%
}

\@Glsuserv@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\Glsentryuserv{#2}}{#2}#3}%
}

\@GLSuserv@
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuservaccessdisplay{\glsuppercase{\glsentryuserv{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

\@glsuservi@
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\glsentryuservi{#2}}{#2}#3}%
}

\@Glsuservi@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\Glsentryuservi{#2}}{#2}#3}%
}

\@GLSuservi@
\def\@GLSuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuserviaccessdisplay{\glsuppercase{\glsentryuservi{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

```

6.3 Displaying the Glossary

We need to redefine the way the glossary entries are formatted to include the accessibility support. The predefined glossary styles use `\glossentryname`, `\glossentrydesc` and `\glossentrysymbol`, but we need to provide compatibility with earlier versions in case users have defined their own styles using `\accsuppglossaryentryfield` and `\accsuppglossarysubentryfield`.

Now redefine `\glossentryname`, `\glossentrydesc` and `\glossentrysymbol` etc so they use the accessibility stuff.

```

\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\glsentryname{#1}}{#1}}%
  }%
}

```

```

\renewcommand*\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\Glsentryname{#1}}{#1}}%
  }%
}

\renewcommand*\glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}

\renewcommand*\Glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}

\renewcommand*\glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
  }%
}

\renewcommand*\Glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
  }%
}

```

\accsuppglossaryentryfield

```

\newcommand*\accsuppglossaryentryfield}[5]{%
  \glossaryentryfield{#1}%
  {\glsnameaccessdisplay{#2}{#1}}%
  {\glsdescriptionaccessdisplay{#3}{#1}}%
  {\glssymbolaccessdisplay{#4}{#1}}{#5}%
}

```

\accsuppglossarysubentryfield

```

\newcommand*\accsuppglossarysubentryfield}[6]{%
  \glossarysubentryfield{#1}{#2}%
  {\glsnameaccessdisplay{#3}{#2}}%
  {\glsdescriptionaccessdisplay{#4}{#2}}%
  {\glssymbolaccessdisplay{#5}{#2}}{#6}%
}

```

6.4 Acronyms

Redefine acronym styles provided by glossaries:

`long-short` $\langle long \rangle$ ($\langle short \rangle$) acronym style.

```
\renewacronymstyle{long-short}%  
{%
```

Check for long form in case this is a mixed glossary.

```
\ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%  
}%  
{%  
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%  
\renewcommand*{\genacrfullformat}[2]{%  
\glslongaccessdisplay{\glsentrylong{##1}}{##1}##2\space  
\glsshortaccessdisplay  
\protect\firstacronymfont{\glsentryshort{##1}}{##1})%  
}%  
\renewcommand*{\Genacrfullformat}[2]{%  
\glslongaccessdisplay{\Glsentrylong{##1}}{##1}##2\space  
\glsshortaccessdisplay  
\protect\firstacronymfont{\glsentryshort{##1}}{##1})%  
}%  
\renewcommand*{\genplacrfullformat}[2]{%  
\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}##2\space  
\glsshortpluralaccessdisplay  
\protect\firstacronymfont{\glsentryshortpl{##1}}{##1})%  
}%  
\renewcommand*{\Genplacrfullformat}[2]{%  
\glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}##2\space  
\glsshortpluralaccessdisplay  
\protect\firstacronymfont{\glsentryshortpl{##1}}{##1})%  
}%  
\renewcommand*{\acronymentry}[1]{%  
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}}  
\renewcommand*{\acronymsort}[2]{##1}%  
\renewcommand*{\acronymfont}[1]{##1}%  
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%  
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%  
}
```

`short-long` $\langle short \rangle$ ($\langle long \rangle$) acronym style.

```
\renewacronymstyle{short-long}%  
{%
```

Check for long form in case this is a mixed glossary.

```
\ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%  
}%  
{%  
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%  
\renewcommand*{\genacrfullformat}[2]{%
```

```

\glsshortaccessdisplay
  {\protect\firstacronymfont{\glentryshort{##1}}}{##1}##2\space
(\glslongaccessdisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
  \glsshortaccessdisplay
    {\protect\firstacronymfont{\Glentryshort{##1}}}{##1}##2\space
(\glslongaccessdisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
  \glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\glentryshortpl{##1}}}{##1}##2\space
(\glslongpluralaccessdisplay
  {\glentrylongpl{##1}}{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\Glentryshortpl{##1}}}{##1}##2\space
(\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}

```

`long-short-desc` *⟨long⟩* (*{⟨short⟩}*) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{long-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1})%
}

```

`long-sc-short-desc` *⟨long⟩* (`\textsc{⟨short⟩}`) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{long-sc-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sc-short}%
}%

```

```

    {%
      \GlsUseAcrStyleDefs{long-sc-short}%
      \renewcommand*\GenericAcronymFields{}%
      \renewcommand*\acronymsort}[2]{##2}%
      \renewcommand*\acronymentry}[1]{%
        \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
    }

long-sm-short-desc <long> (\textsmaller{<short>}) acronym style that has an accompanying de-
scription (which the user needs to supply).
  \renewacronymstyle{long-sm-short-desc}%
  {%
    \GlsUseAcrEntryDispStyle{long-sm-short}%
  }%
  {%
    \GlsUseAcrStyleDefs{long-sm-short}%
    \renewcommand*\GenericAcronymFields{}%
    \renewcommand*\acronymsort}[2]{##2}%
    \renewcommand*\acronymentry}[1]{%
      \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
      (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
  }

short-long-desc <short> ({<long>}) acronym style that has an accompanying description (which
the user needs to supply).
  \renewacronymstyle{short-long-desc}%
  {%
    \GlsUseAcrEntryDispStyle{short-long}%
  }%
  {%
    \GlsUseAcrStyleDefs{short-long}%
    \renewcommand*\GenericAcronymFields{}%
    \renewcommand*\acronymsort}[2]{##2}%
    \renewcommand*\acronymentry}[1]{%
      \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
      (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
  }

sc-short-long-desc <long> (\textsc{<short>}) acronym style that has an accompanying description
(which the user needs to supply).
  \renewacronymstyle{sc-short-long-desc}%
  {%
    \GlsUseAcrEntryDispStyle{sc-short-long}%
  }%
  {%
    \GlsUseAcrStyleDefs{sc-short-long}%
    \renewcommand*\GenericAcronymFields{}%
    \renewcommand*\acronymsort}[2]{##2}%
  }

```

```

\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}

```

sm-short-long-desc *⟨long⟩* (`\textsmaller{⟨short⟩}`) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}

```

dua *⟨long⟩* only acronym style.

```

\renewacronymstyle{dua}%
{%

```

Check for long form in case this is a mixed glossary.

```

\ifdefempty\glscustomtext
{%
  \ifglsahaslong{\glslabel}%
  {%
    \glsifplural
    {%

```

Plural form:

```

\gls caps case
{%

```

Plural form, don't adjust case:

```

\gls long plural access display {\glsentrylongpl{\glslabel}}{\glslabel}%
\gls insert
}%
{%

```

Plural form, make first letter upper case:

```

\gls long plural access display {\Glsentrylongpl{\glslabel}}{\glslabel}%
\gls insert
}%
{%

```

Plural form, all caps:

```

\gls long plural access display
{\glsupercase{\glsentrylongpl{\glslabel}}}{\glslabel}%

```



```

        \glsuppercase{\glsinsert}%
    }%
}%
{%-
Singular form
        \glscapscase
    {%-
Singular form, don't adjust case:
        \glslongaccessdisplay{\glsentrylong{\glslabel}}{\glslabel}\glsinsert
    }%
    {%-
Subsequent singular form, make first letter upper case:
        \glslongaccessdisplay{\Glsentrylong{\glslabel}}{\glslabel}\glsinsert
    }%
    {%-
Subsequent singular form, all caps:
        \glslongaccessdisplay
        {\glsuppercase
         {\glsentrylong{\glslabel}\glsinsert}}{\glslabel}%
        \glsuppercase{\glsinsert}%
    }%
    }%
}%
{%-
Not an acronym:
        \glsgenentryfmt
    }%
}%
{\glscustomtext\glsinsert}%
}%
{%-
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%-
    \glslink[##1]{##2}{%-
        \glslongaccessdisplay{\glsentrylong{##2}}{##2}##3\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}{##2})}}%
\renewcommand*{\Acrfullfmt}[3]{%-
    \glslink[##1]{##2}{%-
        \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%-
    \glslink[##1]{##2}{%-
        \glslongaccessdisplay
        {\glsuppercase{\glsentrylong{##2}}{##2}##3\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}{##2})}}%
\renewcommand*{\acrfullplfmt}[3]{%-
    \glslink[##1]{##2}{%-

```

```

\glslongpluralaccessdisplay
  {\glsentrylongpl{##2}}{##2}##3\space
(\glsshortpluralaccessdisplay
  {\acronymfont{\glsentryshortpl{##2}}{##2}})%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\Glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
      {\acronymfont{\glsentryshortpl{##2}}{##2}})%
  }%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\glsupercase{\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
      {\acronymfont{\glsentryshortpl{##2}}{##2}})%
  }%
\renewcommand*{\glsentryfull}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
  \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}\space
  (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space
  (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}{##1}})%
}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}

```

dua-desc *<long>* only acronym style with user-supplied description.

```

\renewacronymstyle{dua-desc}%
{%
  \GlsUseAcrEntryDispStyle{dua}%
}%
{%
  \GlsUseAcrStyleDefs{dua}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\acronymfont{\glsentrylong{##1}}{##1}}%
\renewcommand*{\acronymsort}[2]{##2}%

```

}%

footnote *<short>*\footnote{*<long>*} acronym style.

```
\renewacronymstyle{footnote}%
```

{%

Check for long form in case this is a mixed glossary.

```
\ifglshaslong{\glslabel}{\glsгенacfmt}{\glsгенentryfmt}%
```

}%

{%

```
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
```

Need to ensure hyperlinks are switched off on first use:

```
\glshyperfirstfalse
```

```
\renewcommand*{\genacrfullformat}[2]{%
```

```
\glsshortaccessdisplay
```

```
{\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2%
```

```
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}}{##1}}%
```

}%

```
\renewcommand*{\Genacrfullformat}[2]{%
```

```
\glsshortaccessdisplay
```

```
{\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%
```

```
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}}{##1}}%
```

}%

```
\renewcommand*{\genplacrfullformat}[2]{%
```

```
\glsshortpluralaccessdisplay
```

```
{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2%
```

```
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%
```

}%

```
\renewcommand*{\Genplacrfullformat}[2]{%
```

```
\glsshortpluralaccessdisplay
```

```
{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%
```

```
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%
```

}%

```
\renewcommand*{\acronymentry}[1]{%
```

```
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
```

```
\renewcommand*{\acronymsort}[2]{##1}%
```

```
\renewcommand*{\acronymfont}[1]{##1}%
```

```
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
```

Don't use footnotes for \acrfull:

```
\renewcommand*{\acrfullfmt}[3]{%
```

```
\glslink[##1]{##2}{%
```

```
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}##3\space
```

```
(\glslongaccessdisplay{\glsentrylong{##2}}{##2}))}%
```

```
\renewcommand*{\Acrfullfmt}[3]{%
```

```
\glslink[##1]{##2}{%
```

```
\glsshortaccessdisplay{\acronymfont{\Glsentryshort{##2}}}{##2}##3\space
```

```
(\glslongaccessdisplay{\glsentrylong{##2}}{##2}))}%
```

```
\renewcommand*{\ACRfullfmt}[3]{%
```

```
\glslink[##1]{##2}{%
```

```

\glsshortaccessdisplay
  {\glsupercase
   {\acronymfont{\glsentryshort{##2}}{##2}##3\space
   (\glslongaccessdisplay{\glsentrylong{##2}}{##2})}}}%
\renewcommand*\acrfullplfmt}[3]{%
\glslink[##1]{##2}{%
\glsshortpluralaccessdisplay
  {\acronymfont{\glsentryshortpl{##2}}{##2}##3\space
  (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}}%
\renewcommand*\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{%
\glsshortpluralaccessdisplay
  {\acronymfont{\Glsentryshortpl{##2}}{##2}##3\space
  (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}}%
\renewcommand*\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%
\glsshortpluralaccessdisplay
  {\glsupercase
   {\acronymfont{\glsentryshortpl{##2}}{##2}##3\space
   (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}}}}%

```

Similarly for `\glsentryfull` etc:

```

\renewcommand*\glsentryfull}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}\space
(\glslongaccessdisplay{\glsentrylong{##1}}{##1})}%
\renewcommand*\Glsentryfull}[1]{%
\glsshortaccessdisplay{\acronymfont{\Glsentryshort{##1}}{##1}\space
(\glslongaccessdisplay{\glsentrylong{##1}}{##1})}%
\renewcommand*\glsentryfullpl}[1]{%
\glsshortpluralaccessdisplay
  {\acronymfont{\glsentryshortpl{##1}}{##1}\space
  (\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1})}%
\renewcommand*\Glsentryfullpl}[1]{%
\glsshortpluralaccessdisplay
  {\acronymfont{\Glsentryshortpl{##1}}{##1}\space
  (\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1})}%
}

```

`footnote-sc \textsc{<short>}\footnote{<long>}` acronym style.

```

\renewacronymstyle{footnote-sc}%
{%
\GlsUseAcrEntryDispStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}}
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*\acrpluralsuffix{\glsstextup{\glspluralsuffix}}%
}%

```

footnote-sm \textsmaller{<short>}\footnote{<long>} acronym style.

```
\renewacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}%
```

footnote-desc <short>\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
\renewacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1))}%
  }
}
```

footnote-sc-desc \textsc{<short>}\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
\renewacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1))}%
  }
}
```

footnote-sm-desc \textsmaller{<short>}\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
\renewacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}%
```

```
{%
\GlsUseAcrStyleDefs{footnote-sm}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}
```

```
\glsdefaultshortaccess{<long>}{<short>}
```

`\glsdefaultshortaccess`

Default shortaccess value.

```
\newcommand*\glsdefaultshortaccess}[2]{#1}
```

Use `\newacronymhook` to modify the key list to set the access text to the long version by default.

```
\renewcommand*\newacronymhook{%
\protected@edef\gls@keylist{%
shortaccess={\glsdefaultshortaccess{the\glslongtok}{the\glsshorttok}},%
shortpluralaccess={\glsdefaultshortaccess{the\glslongtok}{the\glsshorttok}},%
\the\glskeylisttok}%
\expandafter\glskeylisttok\expandafter{\@gls@keylist}%
}
```

`\DefaultNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\DescriptionFootnoteNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\DescriptionNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\FootnoteNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\SmallNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

The following are kept for compatibility with versions before 3.0:

`\glsshortaccesskey`

```
\newcommand*\glsshortaccesskey{\glsshortkey access}%
```

`\glsshortpluralaccesskey`

```
\newcommand*\glsshortpluralaccesskey{\glsshortpluralkey access}%
```

`\glslongaccesskey`

```
\newcommand*\glslongaccesskey{\glslongkey access}%
```

```
\glslongpluralaccesskey
    \newcommand*{\glslongpluralaccesskey}{\glslongpluralkey access}%
```

6.5 Debugging Commands

```
\showglongnameaccess
    \newcommand*{\showglongnameaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@access\endcsname
    }
```

```
\showglongtextaccess
    \newcommand*{\showglongtextaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@textaccess\endcsname
    }
```

```
\showglongpluralaccess
    \newcommand*{\showglongpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@pluralaccess\endcsname
    }
```

```
\showglongfirstaccess
    \newcommand*{\showglongfirstaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@firstaccess\endcsname
    }
```

```
\showglongfirstpluralaccess
    \newcommand*{\showglongfirstpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpluralaccess\endcsname
    }
```

```
\showglongsymbolaccess
    \newcommand*{\showglongsymbolaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolaccess\endcsname
    }
```

```
\showglongsymbolpluralaccess
    \newcommand*{\showglongsymbolpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolpluralaccess\endcsname
    }
```

```
\showglongdescaccess
    \newcommand*{\showglongdescaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@descaccess\endcsname
    }
```

```
\showglongdescpluralaccess
    \newcommand*{\showglongdescpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@descpluralaccess\endcsname
    }
```

`\showgloshortaccess`

```
\newcommand*{\showgloshortaccess}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@shortaccess\endcsname
}
```

`\showgloshortpluralaccess`

```
\newcommand*{\showgloshortpluralaccess}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@shortpluralaccess\endcsname
}
```

`\showglolongaccess`

```
\newcommand*{\showglolongaccess}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@longaccess\endcsname
}
```

`\showglolongpluralaccess`

```
\newcommand*{\showglolongpluralaccess}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@longpluralaccess\endcsname
}
```

7 Backwards Compatibility

7.1 glossaries-compatible-207

Provides compatibility with version 2.07 and below. This uses original glossaries indy and makeindex formatting, so can be used with old documents that had customized style files, but hyperlinks may not work properly.

Now only available with rollback.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossaries-compatible-207-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-compatible-207-2021-11-01.sty}
\DeclareCurrentRelease{v4.56}{2025-03-03}
```

Identify package.

```
\ProvidesPackage{glossaries-compatible-207}[2025/03/03 v4.56 (NLCT)]
```

Trigger error and that's it.

```
\PackageError{glossaries}
{glossaries-compatible-207.sty no longer available. Use rollback if you need it}%
{Try \string\usepackage{glossaries}[=v4.46]}
```

7.2 glossaries-compatible-307

Provides compatibility with version 3.07 and below.

Now only available with rollback.

```
\NeedsTeXFormat{LaTeX2e}
```


Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossaries-compatible-307-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-compatible-307-2021-11-01.sty}
\DeclareCurrentRelease{v4.56}{2025-03-03}
```

Identify package.

```
\ProvidesPackage{glossaries-compatible-307}[2025/03/03 v4.56 (NLCT)]
```

Trigger error and that's it.

```
\PackageError{glossaries}
{glossaries-compatible-307.sty no longer available. Use rollback if you need it}%
{Try \string\usepackage{glossaries} [=v4.46]}
```

8 Rollback

8.1 Rollback v4.46 (glossaries-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{ifthen}
\RequirePackage{xkeyval}[2006/11/18]
\RequirePackage{mfirstuc}
\RequirePackage{textcase}
\renewcommand*{\mfirstucMakeUppercase}{\MakeTextUppercase}%
\RequirePackage{xfor}
\RequirePackage{datatool-base}
\RequirePackage{amsgen}
\RequirePackage{etoolbox}
\newif\if@gls@docloaded
\@ifpackageloaded{doc}%
{
  \@gls@docloadedtrue
}%
{
  \@gls@docloadedfalse
}
\if@gls@docloaded
  \let\glsorg@theglossary\theglossary
  \let\glsorg@endtheglossary\endtheglossary
  \let\glsorg@PrintChanges\PrintChanges
  \renewcommand{\PrintChanges}{%
    \begingroup
      \let\theglossary\glsorg@theglossary
      \let\endtheglossary\glsorg@endtheglossary
      \glsorg@PrintChanges
    \endgroup
  }
\fi
```

```

\newif\if@gls@debug
\define@choicekey{glossaries.sty}{debug}[\gls@debug@val\gls@debug@nr]%
{true,false,showtargets,showaccsupp}[true]{%
  \ifcase\gls@debug@nr\relax
    % debug=true
    \@gls@debugtrue
    \renewcommand*\GlossariesWarning}[1]{%
      \PackageWarning{glossaries}{##1}%
    }%
    \renewcommand*\GlossariesWarningNoLine}[1]{%
      \PackageWarningNoLine{glossaries}{##1}%
    }%
    \let\@glsshowtarget\@gobble
    \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
  \or
    % debug=false
    \@gls@debugfalse
    \let\@glsshowtarget\@gobble
    \let\@glsshowaccsupp\@gobblethree
    \PackageInfo{glossaries}{debug mode OFF}%
  \or
    % debug=showtargets
    \@gls@debugtrue
    \renewcommand*\GlossariesWarning}[1]{%
      \PackageWarning{glossaries}{##1}%
    }%
    \renewcommand*\GlossariesWarningNoLine}[1]{%
      \PackageWarningNoLine{glossaries}{##1}%
    }%
    \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
    \renewcommand{\@glsshowtarget}{\@glsshowtarget}%
  \or
    % debug=showaccsupp
    \@gls@debugtrue
    \renewcommand*\GlossariesWarning}[1]{%
      \PackageWarning{glossaries}{##1}%
    }%
    \renewcommand*\GlossariesWarningNoLine}[1]{%
      \PackageWarningNoLine{glossaries}{##1}%
    }%
    \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
    \renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
  \fi
}
\newcommand*\glsshowtarget}[1]{%
  \ifmmode
    \nfss@text{\glsshowtargetfont [#1]}%
  \else
    \ifinner
      {\glsshowtargetfont [#1]}%
    \fi
  \fi
}

```

```

\else
  \glsshowtargetouter{#1}%
\fi
\fi
}
\newcommand{\glsshowtargetouter}[1]{%
  \glsshowtargetsymbolsymbol\marginpar{\glsshowtargetsymbolsymbol\glsshowtargetfont #1}}
\newcommand{\glsshowtargetsymbolsymbol}{\tiny$\triangleright$}
\newcommand{\glsshowtargetfont}{\ttfamily\footnotesize}
\newcommand{\@glsshowtarget}[1]{%
\newrobustcmd*{\@glsshowtarget}[1]{%
  \begingroup
  \protected@edef\@gls@tmp{#1}%
  \@onelevel@sanitize\@gls@tmp
  \expandafter\glsshowtarget\expandafter{\@gls@tmp}%
  \endgroup
}
\newcommand{\@glsshowaccsupp}[3]{%
\newrobustcmd*{\@glsshowaccsupp}[3]{%
  \ifstrempy{#1}%
  {\@glsshowtarget{/#2 (#3)}}%
  {\@glsshowtarget{/#2 (#3) [#1]}}%
}
\newcommand{\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries\space (or move
  \string\newglossaryentry\space
  definitions into the preamble)}}%
{You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key. It may
  be that the 'see' key has been written to the .glsdefs
  file from the previous run, in which case you need to
  move your definitions
  to the preamble if you don't want to use
  \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoinde}x{%
  [\gls@seenoinde@val\gls@seenoinde@nr]{error,warn,ignore}{%
  \ifcase\gls@seenoinde@nr
  \renewcommand{\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries}%
  {You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key}}%
  }%
}

```

```

\or
  \renewcommand*\@gls@see@noindex}{%
    \GlossariesWarning{'\gls@xr@key' key ignored}%
  }%
\or
  \renewcommand*\@gls@see@noindex}{}%
\fi
}
\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}%
  {\newcommand*\@@glossarysec}{section}}%
  {\newcommand*\@@glossarysec}{chapter}}
\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
subsection,subsection,paragraph,subparagraph}[section]{%
  \renewcommand*\@@glossarysec{#1}}
\newcommand*\@@glossarysecstar}{*}%
\newcommand*\@@glossaryseclabel}{%
\newcommand*\glsautoprefix{}%
\define@choicekey{glossaries.sty}{numberedsection}%
  [\gls@numberedsection@val\gls@numberedsection@nr]{%
false,nolabel,autolabel,nameref}[nolabel]{%
  \ifcase\gls@numberedsection@nr\relax
    \renewcommand*\@@glossarysecstar}{*}%
    \renewcommand*\@@glossaryseclabel}{%
  \or
    \renewcommand*\@@glossarysecstar}{}%
    \renewcommand*\@@glossaryseclabel}{%
  \or
    \renewcommand*\@@glossarysecstar}{}%
    \renewcommand*\@@glossaryseclabel}{%
    \label{\glsautoprefix\@glo@type}}%
  \or
    \renewcommand*\@@glossarysecstar}{*}%
    \renewcommand*\@@glossaryseclabel}{%
    \protected@edef\@currentlabelname{\glossarytoctitle}%
    \label{\glsautoprefix\@glo@type}}%
  \fi
}
\@ifpackageloaded{classicthesis}
{\newcommand*\@glossary@default@style}{index}}
{\newcommand*\@glossary@default@style}{list}}
\define@key{glossaries.sty}{style}{%
  \def\@glossary@default@style{#1}%
}
\newcommand*\@gls@declareoption}[2]{%
  \DeclareOptionX{#1}{#2}%
  \DeclareOption{#1}{#2}%
}
\newcommand*\glossaryentrynumbers}[1]{#1\gls@save@numberlist{#1}}

```

```

\@gls@declareoption{nonumberlist}{%
  \renewcommand*\glossaryentrynumbers}[1]{\gls@save@numberlist{#1}}%
}
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{%
\gls@savenumberlistfalse
\newcommand*\glo@seeautonumberlist{}
\@gls@declareoption{seeautonumberlist}{%
  \renewcommand*\glo@seeautonumberlist}{%
    \def\glo@prefix{\glsnextpages}%
  }%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{%
\gls@esclocationstrue
\newcommand*\gls@loadlong{\RequirePackage{glossary-long}[=v4.46]}
\@gls@declareoption{nolong}{\renewcommand*\gls@loadlong{}}
\IfFileExists{supertabular.sty}{%
  \newcommand*\gls@loadsuper{\RequirePackage{glossary-super}[=v4.46]}}{%
  \newcommand*\gls@loadsuper{}}
\@gls@declareoption{nosuper}{\renewcommand*\gls@loadsuper{}}
\newcommand*\gls@loadlist{\RequirePackage{glossary-list}[=v4.46]}
\@gls@declareoption{nolist}{%
  \renewcommand*\gls@loadlist}{%
    \ifdefstring{\glossary@default@style}{list}%
    {\let\glossary@default@style\relax}%
  }%
}
\newcommand*\gls@loadtree{\RequirePackage{glossary-tree}[=v4.46]}
\@gls@declareoption{notree}{\renewcommand*\gls@loadtree{}}
\@gls@declareoption{nostyles}{%
  \renewcommand*\gls@loadlong}{%
  \renewcommand*\gls@loadsuper}{%
  \renewcommand*\gls@loadlist}{%
  \renewcommand*\gls@loadtree}{%
  \let\glossary@default@style\relax
}
\newcommand*\gls@postdescription}{%
  \ifglsnopostdot\else.\spacefactor\sfcode'\. \fi
}
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{%
\gls@nopostdotfalse
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{%
\gls@nogroupskipfalse
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{%
\@ifclassloaded{memoir}
{%
  \glsucmarktrue
}%
}
}
\glsucmarkfalse

```

```

}
\newcommand*{\@gls@define@glossaryentrycounter}{%
  \ifglentrycounter
    \ifundef\c@glossaryentry
      {%
        \ifx\@gls@counterwithin\@empty
          \newcounter{glossaryentry}%
        \else
          \newcounter{glossaryentry}[\@gls@counterwithin]%
        \fi
        \def\theHglossaryentry{\currentglossary.\theglossaryentry}%
      }%
    }%
  \fi
}
\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\glentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
  \renewcommand*{\@gls@counterwithin}{#1}%
  \glentrycountertrue
  \@gls@define@glossaryentrycounter
}
\newcommand*{\@gls@counterwithin}{}
\newcommand{\@gls@define@glossarysubentrycounter}{%
  \ifundef\c@glossarysubentry
    {%
      \ifglssubentrycounter
        \ifglentrycounter
          \newcounter{glossarysubentry}[glossaryentry]%
        \else
          \newcounter{glossarysubentry}%
        \fi
        \def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
      \fi
    }%
  }%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
\glssubentrycounterfalse
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none}{%
  \renewcommand*{\@glo@default@sorttype}{#1}%
  \csname @gls@setupsort@#1\endcsname
}
\newcommand*{\glsprestandardsort}[3]{%
  \glsdosanitizesort
}
\newcommand*{\@glo@check@sortallowed}[1]{}
\newcommand*{\@gls@setupsort@standard}{%
  \def\do@glo@storeentry{\@glo@storeentry}%
}

```

```

\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
  \ifx\@glo@sort\@glsdefaultsort
    \let\@glo@sort\@glo@name
  \fi
  \let\glsdosanitizesort\@gls@sanitizesort
  \glsprestandardsort{\@glo@sort}{##1}{##2}%
  \expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\@gls@setupsort@standard
\newcommand*\glsortnumberfmt[1]{%
  \ifnum#1<100000 0\fi
  \ifnum#1<10000 0\fi
  \ifnum#1<1000 0\fi
  \ifnum#1<100 0\fi
  \ifnum#1<10 0\fi
  \number#1%
}
\newcommand*\@gls@setupsort@def{%
\def\do@glo@storeentry{\@glo@storeentry}%
\def\@gls@defsortcount##1{%
  \expandafter\global
  \expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
  \ifcsundef{glossary@##1@sortcount}%
  {\@gls@defsortcount{##1}}%
  {}%
  \expandafter\global\expandafter
  \advance\csname glossary@##1@sortcount\endcsname by 1\relax
  \expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
    \expandafter\glsortnumberfmt
    {\csname glossary@##1@sortcount\endcsname}}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*\@gls@setupsort@use{%
\let\do@glo@storeentry\@gobble
\def\@gls@defsortcount##1{%
  \expandafter\global
  \expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
  \expandafter\gdef\csname glo@##2@sort\endcsname{}%
}%
\def\@gls@setsort##1{%

```

```

\edef\@glo@parent{\csname glo@##1@parent\endcsname}%
\ifx\@glo@parent\@empty
\else
\expandafter\@gls@setsort\expandafter{\@glo@parent}%
\fi
\edef\@glo@type{\csname glo@##1@type\endcsname}%
\edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
\ifx\@gls@tmp\@empty
\expandafter\global\expandafter
\advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
\expandafter\protected\edef\csname glo@##1@sort\endcsname{%
\expandafter\gls@sortnumberfmt
{\csname glossary@\@glo@type @sortcount\endcsname}}%
\@glo@storeentry{##1}%
\fi
}%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*{\@gls@setupsort@none}{%
\def\do@glo@storeentry##1{%
\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
\expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort
}%
\def\@gls@setsort##1{%
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
{Option sort=none not allowed with \string##1}%
{(Use sort=def instead)}}%
}
\newcommand*{\glsdefmain}{%
\if@gls@docloaded
\newglossary[glg2]{main}{gls2}{glo2}{\glossaryname}%
\else
\newglossary{main}{gls}{glo}{\glossaryname}%
\fi
\newcommand*{\gls@tr@set@main@toctitle}{%
\translatelet{\glossarytoctitle}{Glossary}%
}%
}
\newcommand*{\glsdefaulttype}{main}
\newcommand*{\acronymtype}{\glsdefaulttype}
\@gls@declareoption{nomain}{%
\let\glsdefaulttype\relax
\renewcommand*{\glsdefmain}{}%
}
\define@boolkey{glossaries.sty}{gls}{acronym}[true]{%
\ifglsacronym
\renewcommand{\@gls@do@acronymsdef}{%
\DeclareAcronymList{acronym}%
\newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
}

```



```

        \renewcommand*\acronymtype}{acronym}%
        \newcommand*\gls@tr@set@acronym@toctitle}{%
            \translatelet{\glossarytoctitle}{Acronyms}%
        }%
    }%
\else
    \let\gls@do@acronymsdef\relax
\fi
}
\AtBeginDocument{%
    \ifglsacronym
        \ifbool{glscompatible-3.07}%
            {}%
            {%
                \providecommand*\printacronyms}[1][{}]{%
                    \printglossary[type=\acronymtype,#1]}%
            }%
        \fi
    }
\newcommand*\gls@do@acronymsdef{}
\gls@declareoption{acronyms}{%
    \glsacronymtrue
    \def\gls@do@acronymsdef{%
        \DeclareAcronymList{acronym}%
        \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
        \renewcommand*\acronymtype}{acronym}%
        \newcommand*\gls@tr@set@acronym@toctitle}{%
            \translatelet{\glossarytoctitle}{Acronyms}%
        }%
    }%
}
\newcommand*\glsacronymlists{}
\newcommand*\@addtoacronymlists}[1]{%
    \ifx\glsacronymlists\@empty
        \protected@xdef\glsacronymlists{#1}%
    \else
        \protected@xdef\glsacronymlists{\glsacronymlists,#1}%
    \fi
}
\newcommand*\DeclareAcronymList}[1]{%
    \glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}
\newcommand*\glsIfListOfAcronyms}[1]{%
    \edef\do@gls@islistofacronyms{%
        \noexpand\gls@islistofacronyms{#1}{\glsacronymlists}}%
    \do@gls@islistofacronyms
}
\newcommand*\@gls@islistofacronyms}[4]{%
    \def\gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%
        \def\@gls@before{##1}\def\@gls@after{##2}}%
}

```

```

\gls@islistofacronyms,#2,#1,\@nil\end@gls@islistofacronyms
\ifx\@gls@after\@nnil
  #4%
\else
  #3%
\fi
}
\newif\if@gls@isacronymlist
\newcommand*\@gls@checkisacronymlist}[1]{%
  \glsIfListOfAcronyms{#1}%
  {\@gls@isacronymlisttrue}{\@gls@isacronymlistfalse}%
}
\newcommand*\@SetAcronymLists}[1]{%
  \renewcommand*\@gls@acronymlists{#1}%
}
\define@key{glossaries.sty}{acronymlists}{%
  \DeclareAcronymList{#1}%
}
\newcommand*\@gls@counter{page}
\define@key{glossaries.sty}{counter}{%
  \renewcommand*\@gls@counter{#1}%
}
\newcommand*\@gls@nohyperlist{}
\newcommand*\@GlsDeclareNoHyperList}[1]{%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*\@gls@nohyperlist{#1}%
  }%
  {%
    \appto\@gls@nohyperlist{,#1}%
  }%
}
\define@key{glossaries.sty}{nohypertypes}{%
  \GlsDeclareNoHyperList{#1}%
}
\newcommand*\@GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{#1}%
}
\newcommand*\@GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{#1}%
}
\newcommand*\@glosortentrieswarning{%
  \typeout{Using TeX to sort glossary entries---this may
  take a while}%
}
\@gls@declareoption{nowarn}{%
  \if@gls@debug
    \GlossariesWarning{Warnings can't be suppressed in debug mode}%
  \else
    \renewcommand*\@GlossariesWarning}[1]{%

```

```

        \renewcommand*\GlossariesWarningNoLine}[1]{}%
        \renewcommand*\glosortentrieswarning}{}%
        \renewcommand*\@gls@missinglang@warn}[2]{}%
    \fi
}
\newcommand*\@gls@missinglang@warn}[2]{%
    \PackageWarningNoLine{glossaries}%
    {No language module detected for ‘#1’.\MessageBreak
    Language modules need to be installed separately.\MessageBreak
    Please check on CTAN for a bundle called\MessageBreak
    ‘glossaries-#2’ or similar}%
}
\@gls@declareoption{no-langwarn}{%
    \renewcommand*\@gls@missinglang@warn}[2]{}%
}
\newcommand*\@gls@warnonglossdefined}{%
    \GlossariesWarning{Overriding \string\printglossary}%
}
\newcommand*\@gls@warnontheglossdefined}{%
    \GlossariesWarning{Overriding ‘theglossary’ environment}%
}
\@gls@declareoption{no-undefwarn}{%
    \renewcommand*\@gls@warnonglossdefined}{}%
    \renewcommand*\@gls@warnontheglossdefined}{}%
}
\newcommand*\@gls@sanitizedesc}{%
}
\newcommand*\glssetexpandfield}[1]{%
    \csdef{gls@assign@#1@field}##1##2{%
        \@gls@expand@field{##1}{#1}{##2}%
    }%
}
\newcommand*\glssetnoexpandfield}[1]{%
    \csdef{gls@assign@#1@field}##1##2{%
        \@gls@noexpand@field{##1}{#1}{##2}%
    }%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}
\glssetnoexpandfield{desplural}
\newcommand*\@gls@sanitizename}{%
\glssetnoexpandfield{name}
\newcommand*\@gls@sanitizesymbol}{%
\glssetnoexpandfield{symbol}
\glssetnoexpandfield{symbolplural}
\newcommand*\@gls@sanitizesort}{%
    \ifgls@sanitizesort
        \@gls@sanitizesort
    \else
        \@gls@nosanitizesort

```

```

\fi
}
\newcommand*{\@@gls@sanitizesort{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@@gls@nosanitizesort}{%}
\newcommand*{\@gls@noidx@sanitizesort{%
  \ifdefvoid\@glo@sort
  }{%
  {%
  \expandafter\@@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
  \def\@glo@sort{#1#2}%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@@gls@noidx@nosanitizesort}{%
  \ifdefvoid\@glo@sort
  }{%
  {%
  \expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
  \bgroup
  \glsnoidxstripaccents
  \protected@xdef\@@glo@sort{#1#2}%
  \egroup
  \let\@glo@sort\@@glo@sort
}
\newcommand*{\glsnoidxstripaccents{%
  \let\IeC\@firstofone
  \let\add@accent@\@secondoftwo
  \let\@text@composite@x@\@secondoftwo
  \let\@tabacckludge@\@secondoftwo
  \expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
  \let\'\@firstofone
  \let\'\@firstofone
  \let\^\@firstofone
  \let\\"\@firstofone
  \let\u\@firstofone
  \let\t\@firstofone
  \let\d\@firstofone
  \let\r\@firstofone
  \let\=\@firstofone
  \let\.\@firstofone

```

```

\let~\@firstofone
\let\v\@firstofone
\let\H\@firstofone
\let\c\@firstofone
\let\b\@firstofone
\let\a\@secondoftwo
\def\AE{AE}%
\def\ae{ae}%
\def\OE{OE}%
\def\oe{oe}%
\def\AA{AA}%
\def\aa{aa}%
\def\L{L}%
\def\l{l}%
\def\O{O}%
\def\o{o}%
\def\SS{SS}%
\def\ss{ss}%
\def\th{th}%
\def\TH{TH}%
\def\dh{dh}%
\def\DH{DH}%
}
\@ifl@t@r\fmtversion{2019/10/01}
{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}
\define@boolkey[gls]{sanitize}{description}[true]{%
\GlossariesWarning{sanitize={description} package option deprecated}%
\ifgls@sanitize@description
\glssetnoexpandfield{desc}%
\glssetnoexpandfield{descplural}%
\else
\glssetexpandfield{desc}%
\glssetexpandfield{descplural}%
\fi
}
\define@boolkey[gls]{sanitize}{name}[true]{%
\GlossariesWarning{sanitize={name} package option deprecated}%
\ifgls@sanitize@name
\glssetnoexpandfield{name}%
\else
\glssetexpandfield{name}%
\fi
}
\define@boolkey[gls]{sanitize}{symbol}[true]{%
\GlossariesWarning{sanitize={symbol} package option deprecated}%
\ifgls@sanitize@symbol
\glssetnoexpandfield{symbol}%

```

```

        \glsssetnoexpandfield{symbolplural}%
    \else
        \glsssetexpandfield{symbol}%
        \glsssetexpandfield{symbolplural}%
    \fi
}
\define@boolkey{glossaries.sty}[gls]{sanitizesort}[true]{%
    \ifglsssanitizesort
        \glsssetnoexpandfield{sortvalue}%
        \renewcommand*{\@gls@noidx@setsanitizesort}{%
            \glsssanitizesorttrue
            \glsssetnoexpandfield{sortvalue}%
        }%
    \else
        \glsssetexpandfield{sortvalue}%
        \renewcommand*{\@gls@noidx@setsanitizesort}{%
            \glsssanitizesortfalse
            \glsssetexpandfield{sortvalue}%
        }%
    \fi
}
\glsssanitizesorttrue
\glsssetnoexpandfield{sortvalue}%
\newcommand*{\@gls@noidx@setsanitizesort}{%
    \glsssanitizesortfalse
    \glsssetexpandfield{sortvalue}%
}
\define@choicekey[gls]{sanitize}{sort}{true,false}[true]{%
    \setbool{glsssanitizesort}{#1}%
    \ifglsssanitizesort
        \glsssetnoexpandfield{sortvalue}%
    \else
        \glsssetexpandfield{sortvalue}%
    \fi
}
\GlossariesWarning{sanitize={sort} package option
    deprecated. Use sanitizesort instead}%
}
\define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
    \ifthenelse{\equal{#1}{none}}{%
        {%
            \GlossariesWarning{sanitize package option deprecated}%
            \glsssetexpandfield{name}%
            \glsssetexpandfield{symbol}%
            \glsssetexpandfield{symbolplural}%
            \glsssetexpandfield{desc}%
            \glsssetexpandfield{descplural}%
        }%
    }%
    \setkeys[gls]{sanitize}{#1}%
}%

```

```

}
\newif\ifglstranslate
\newcommand*\@gls@usetranslator{%
  \@ifpackageloaded{polyglossia}%
  {%
    \let\glsifusetranslator\@secondoftwo
  }%
  {%
    \@ifpackageloaded{babel}%
    {%
      \IfFileExists{translator.sty}%
      {%
        \RequirePackage{translator}%
        \let\glsifusetranslator\@firstoftwo
      }%
    }%
  }%
  {}%
}%
}
\newcommand{\glsifusedtranslatordict}[3]{%
  \glsifusetranslator
  {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
  {#3}%
}
\@gls@declareoption{nottranslate}{%
  \glstranslatefalse
  \let\@gls@usetranslator\relax
  \let\glsifusetranslator\@secondoftwo
}
\define@choicekey{glossaries.sty}{translate}%
  [\gls@translate@val\gls@translate@nr]%
  {true,false,babel}[true]%
  {%
    \ifcase\gls@translate@nr\relax
    \glstranslatetrue
    \renewcommand*\@gls@usetranslator{%
      \@ifpackageloaded{polyglossia}%
      {%
        \let\glsifusetranslator\@secondoftwo
      }%
    }%
    \@ifpackageloaded{babel}%
    {%
      \IfFileExists{translator.sty}%
      {%
        \RequirePackage{translator}%
        \let\glsifusetranslator\@firstoftwo
      }%
    }%
  }%
  {}%
}

```

```

        }%
        {}%
    }%
}
\or
\glstranslatefalse
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\or
\glstranslatetrue
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\fi
}
\glstranslatefalse
\let\glsifusetranslator\@secondoftwo
\@ifpackageloaded{translator}%
{%
\glstranslatetrue
\let\glsifusetranslator\@firstoftwo
}%
{%
\@for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
{
\@ifpackageloaded{\gls@thissty}%
{%
\glstranslatetrue
\@endfortrue
}%
{}%
}
}
\define@boolkey{glossaries.sty}[gls]{indexonlyfirst}[true]{}
\glsindexonlyfirstfalse
\define@boolkey{glossaries.sty}[gls]{hyperfirst}[true]{}
\glshyperfirsttrue
\newcommand*{\@gls@setacrstyle}{}
\define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{%
\ifbool{glsacrdescription}%
{}%
{%
\renewcommand*{\@gls@sanitizedesc}{}}%
}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{description}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{%

```



```

\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{dua}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{shortcuts}[true]{}
\newcommand*{\glsorder}{word}
\newcommand*{\@glsorder}[1]{}
\define@choicekey{glossaries.sty}{order}{word,letter}{%
\def\glsorder{#1}}
\newif\ifglsxindy
\glsxindyfalse
\@gls@declareoption{makeindex}{\glsxindyfalse}
\define@boolkey[gls]{xindy}{glsnumbers}[true]{}
\glsxindy@glsnumberstrue
\def\@xdy@main@language{\languagename}%
\define@key[gls]{xindy}{language}{\def\@xdy@main@language{#1}}
\ifcsundef{inputencodingname}{%
\def\gls@codepage{}}{%
\def\gls@codepage{\inputencodingname}
}
\define@key[gls]{xindy}{codepage}{\def\gls@codepage{#1}}
\define@key{glossaries.sty}{xindy}[]{}
\glsxindytrue
\setkeys[gls]{xindy}{#1}%
}
\@gls@declareoption{xindygloss}{%
\glsxindytrue
}
\@gls@declareoption{xindynoglsnumbers}{%
\glsxindytrue
\glsxindy@glsnumbersfalse
}
\providecommand{\@domakeglossaries}[1]{#1}
\@gls@declareoption{disablemakegloss}{%
\ifdefequal\makeglossaries\@no@makeglossaries
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect
(\string\makenoidxglossaries\space has already been used)}}%
}%
{%
\ifdefequal\@makeglossary\@gobble
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect

```

```

        (\string\makeglossaries\space has already been used)}%
    }%
    {%
    \renewcommand{\@domakeglossaries}[1]{%
        \PackageInfo{glossaries}{\string\makeglossaries\space and
        \string\makenoidxglossaries\space have been disabled}%
    }%
    }%
    }%
}
\@gls@declareoption{restoremakegloss}{%
    \ifdefequal\makeglossaries\@no@makeglossaries
    {%
        \GlossariesWarning{Option ‘restoremakegloss’ has no effect
        (\string\makenoidxglossaries\space has already been used)}%
    }%
    {%
        \ifdefequal\@makeglossary\@gobble
        {%
            \GlossariesWarning{Option ‘restoremakegloss’ has no effect
            (\string\makeglossaries\space has already been used)}%
        }%
        {%
            \PackageInfo{glossaries}{\string\makeglossaries\space and
            \string\makenoidxglossaries\space have been restored}%
            \let\@domakeglossaries\@firstofone
        }%
    }%
}
}
\newcommand*\@do@write@glslabels{%
    \AtEndDocument{\@do@write@glslabels}%
    \let\@do@write@glslabels\relax
}
\newcommand*\@do@write@glslabels{%
    \newwrite\@gls@labelsfile
    \immediate\openout\@gls@labelsfile=\jobname.glslabels
    \forallglsentries[\@glo@types,\@ignored@glossaries]{\@glsentry}%
    {\ifdefempty{\@glsentry}{\immediate\write\@gls@labelsfile{\@glsentry}}}%
    \immediate\closeout\@gls@labelsfile
}
\@gls@declareoption{writeglslabels}{\@do@write@glslabels}
\newif\ifglsautomake
\newcommand{\gls@automake@nr}{1}
\define@choicekey{glossaries.sty}{automake}%
    [\gls@automake@val\gls@automake@nr]{true,false,immediate}[true]{%
    \ifnum\gls@automake@nr=1\relax
        \glsautomakefalse
    \else
        \glsautomaketrue
    \fi
}

```

```

\ifglsautomake
  \renewcommand*{\@gls@doautomake}{%
    \PackageError{glossaries}{You must use
    \string\makeglossaries\space with automake=true}
    {%
      Either remove the automake=true setting or
      add \string\makeglossaries\space to your document preamble.%
    }%
  }%
\else
  \renewcommand*{\@gls@doautomake}{}%
\fi
}
\glsautomakefalse
\newcommand*{\@gls@doautomake}{}
\AtEndDocument{\@gls@doautomake}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
  \ifglssavewrites
    \renewcommand*{\glswritefiles}{\@glswritefiles}%
  \else
    \let\glswritefiles\@empty
  \fi
}
\glssavewritesfalse
\let\glswritefiles\@empty
\define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{%
\boolfalse{glscompatible-3.07}
\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
  \ifbool{glscompatible-2.07}%
  {%
    \booltrue{glscompatible-3.07}%
  }%
  {}%
}
\boolfalse{glscompatible-2.07}
\let\gls@original@makeglossary\makeglossary
\let\gls@original@glossary\glossary
\def\makeglossary{%
  \GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is \MessageBreak deprecated. Use \string\makeglossaries\space
instead. If you \MessageBreak need the original definition of
\string\makeglossary\space use \MessageBreak the package options
kernelglossredefs=false (to \MessageBreak restore the former definition of
\string\makeglossary) and \MessageBreak nomain (if the file extensions cause a
conflict)}}%
\makeglossaries
}
\newcommand*{\@gls@override@glossary}[1][main]{%
  \GlossariesWarning{Use of \string\glossary\space with
glossaries.sty is deprecated. \MessageBreak Indexing should be performed

```

```

with the user level \MessageBreak commands, such as \string\gls\space or
\string\glsadd. If you need the \MessageBreak original definition of
\string\glossary\space use the package \MessageBreak options
kernelglossredefs=false (to restore the \MessageBreak former definition of
\string\glossary) and nomain (if the \MessageBreak file extensions cause a
conflict)}%
\gls@glossary{#1}%
}
\if@gls@docloaded
\else
\def\glossary{\@gls@override@glossary}
\fi
\define@choicekey{glossaries.sty}{kernelglossredefs}%
[\gls@debug@val\gls@debug@nr]{true,false,nowarn}[true]%
{%
\ifcase\gls@debug@nr\relax
\def\glossary{\@gls@override@glossary}%
\def\makeglossary{%
\GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is deprecated. Use \string\makeglossaries\space
instead. If you need the original definition of
\string\makeglossary\space use the package options
kernelglossredefs=false (to prevent redefinition of
\string\makeglossary) and nomain (if the file extensions cause a
conflict)}%
\makeglossaries
}%
\or
\let\glossary\gls@original@glossary
\let\makeglossary\gls@original@makeglossary
\or
\def\makeglossary{\makeglossaries}%
\renewcommand*{\@gls@override@glossary}[1][main]{%
\gls@glossary{##1}%
}%
\fi
}
\@gls@declareoption{symbols}{%
\let\@gls@do@symbolsdef\@gls@symbolsdef
}
\newcommand*{\@gls@do@symbolsdef}{}
\newcommand*{\@gls@symbolsdef}{%
\newglossary[slg]{symbols}{sls}{slo}{\glssymbolsgroupname}%
\newcommand*{\printsymbols}[1][\printglossary[type=symbols,##1]}%
\newcommand*{\gls@tr@set@symbols@toctitle}{%
\translatelet{\glossarytoctitle}{Symbols (glossaries)}%
}%
}%
\@gls@declareoption{numbers}{%
\let\@gls@do@numbersdef\@gls@numbersdef

```

```

}
\newcommand*{\@gls@do@numbersdef}{}
\newcommand*{\@gls@numbersdef}{%
  \newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
  \newcommand*{\printnumbers}[1][\printglossary[type=numbers,##1]]{%
  \newcommand*{\gls@tr@set@numbers@toctitle}{%
    \translatelet{\glossarytoctitle}{Numbers (glossaries)}}%
  }%
}%
\@gls@declareoption{index}{%
  \ifx\@gls@do@indexdef\@empty
    \let\@gls@do@indexdef\@gls@indexdef
  \fi
}
\@gls@declareoption{noglossaryindex}{%
  \let\@gls@do@indexdef\relax
}
\newcommand*{\@gls@do@indexdef}{}
\newcommand*{\@gls@indexdef}{%
  \newglossary[ilg]{index}{ind}{idx}{\indexname}%
  \newcommand*{\printindex}[1][\printglossary[type=index,##1]]{%
  \newcommand*{\newterm}[2][%
    \newglossaryentry{##2}{%
      {type={index},name={##2},description={\nopostdesc},##1}}
  \let\@gls@do@indexdef\relax
}%
}%
\@for\CurrentOption :=\@declaredoptions\do{%
  \ifx\CurrentOption\@empty
  \else
    \@expandtwoargs
      \in@ {,\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
    \ifin@
      \@use@ption
      \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
    \fi
  \fi
}
\ProcessOptionsX
\RequirePackage{glossaries-compatible-307}[v4.46]
\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,notranslate,nolong,nosuper,notree,nostyles,%
nomain,noglossaryindex}
\newcommand*{\setupglossaries}[1]{%
  \renewcommand*{\@gls@setacrstyle}{}%
  \ifglsacrshortcuts
    \def\@gls@setupshortcuts{\glsacrshortcutstrue}%
  \else
    \def\@gls@setupshortcuts{%
      \ifglsacrshortcuts

```

```

        \DefineAcronymSynonyms
    \fi
    }%
\fi
\glsacrshortcutsfalse
\let\@gls@do@numbersdef\relax
\let\@gls@do@symbolssdef\relax
\let\@gls@do@indexdef\relax
\let\@gls@do@acronymsdef\relax
\ifglsentrycounter
    \let\@gls@doentrycounterdef\relax
\else
    \let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
\fi
\ifglsentrycounter
    \let\@gls@dosubentrycounterdef\relax
\else
    \let\@gls@dosubentrycounterdef\@gls@define@glossarysubentrycounter
\fi
\setkeys{glossaries.sty}{#1}%
\@gls@setacrstyle
\@gls@setupshortcuts
\@gls@do@acronymsdef
\@gls@do@numbersdef
\@gls@do@symbolssdef
\@gls@do@indexdef
\@gls@doentrycounterdef
\@gls@dosubentrycounterdef
}
\ifthenelse{\equal{\glscounter}{section}}%
{%
    \ifcsundef{chapter}{}%
    {%
        \let\@gls@old@chapter\@chapter
        \def\@chapter[#1]#2{\@gls@old@chapter[#1]}{#2}%
        \ifcsundef{hyperdef}{\hyperdef{section}{\thesection}}{}%
    }%
}%
{}
\newcommand*\@gls@onlypremakeg{}
\newcommand*\@onlypremakeg[1]{%
    \ifx\@gls@onlypremakeg\@empty
        \def\@gls@onlypremakeg{#1}%
    \else
        \expandafter\toks@\expandafter{\@gls@onlypremakeg}%
        \edef\@gls@onlypremakeg{the\toks@,\noexpand#1}%
    \fi
}
\newcommand*\@disable@onlypremakeg{}
\@for\@thiscs:=\@gls@onlypremakeg\do{%

```

```

\expandafter\@disable@premakecs\@thiscs%
}}
\newcommand*\@disable@premakecs}[1]{%
\def#1{\PackageError{glossaries}{\string#1\space may only be
used before \string\makeglossaries}{You can't use
\string#1\space after \string\makeglossaries}}%
}
\providecommand*\glossaryname{Glossary}
\providecommand*\acronymname{Acronyms}
\newcommand*\glssettoctitle}[1]{%
\def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}
\providecommand*\entryname{Notation}
\providecommand*\descriptionname{Description}
\providecommand*\symbolname{Symbol}
\providecommand*\pagelistname{Page List}
\providecommand*\glsymbolsgroupname{Symbols}
\providecommand*\glsnumbersgroupname{Numbers}
\newcommand*\glspluralsuffix}{s}
\newcommand*\glsacrpluralsuffix}{\glspluralsuffix}
\newcommand*\glsupacrpluralsuffix}{\glstextup{\glsacrpluralsuffix}}
\providecommand*\seename}{see}
\providecommand*\andname}{\&}
\newcommand*\RequireGlossariesLang}[1]{%
\@ifundefined{ver@glossaries-#1.1df}{\input{glossaries-#1.1df}}{}%
}
\newcommand*\ProvidesGlossariesLang}[1]{%
\ProvidesFile{glossaries-#1.1df}%
}
\newcommand*\addglossarytocaptions}[1]{}
\ifglstranslate
\RequirePackage{tracklang}
\@gls@usetranslator
\@ifpackageloaded{translator}
{%
\ifboolexpr
{
test {\ifdefstring{\trans@languages}{English}}
and not
test {\ifdefstring{bbl@loaded}{english}}
}
{%
\let\glsifusetranslator\@secondoftwo
}%
{%
\usedictionary{glossaries-dictionary}%
\renewcommand*\addglossarytocaptions}[1]{%
\ifcsundef{captions#1}{}%
{%
\expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
\expandafter\toks@\expandafter{\@gls@tmp

```

```

        \renewcommand*{\glossaryname}{\translate{Glossary}}%
    }%
    \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
  }%
} %
} %
{} %
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
  }%
} %
{} %
\glsifusetranslator
{%
  \renewcommand*{\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*{\glossaryname}{\translate{Glossary}}%
  \renewcommand*{\acronymname}{\translate{Acronyms}}%
  \renewcommand*{\entryname}{\translate{Notation (glossaries)}}%
  \renewcommand*{\descriptionname}{%
    \translate{Description (glossaries)}}%
  \renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}%
  \renewcommand*{\pagelistname}{%
    \translate{Page List (glossaries)}}%
  \renewcommand*{\glsymbolsgroupname}{%
    \translate{Symbols (glossaries)}}%
  \renewcommand*{\glsnumbersgroupname}{%
    \translate{Numbers (glossaries)}}%
} %
\fi
\DeclareRobustCommand*{\nopostdesc}{%
\newcommand*{\@nopostdesc}{%

```



```

\let\org@glspostdescription\glspostdescription
\def\glspostdescription{%
  \let\glspostdescription\org@glspostdescription}%
}
\newcommand*\@no@post@desc{}\nopostdesc}
\newcommand{\glspar}{\par}
\newcommand{\setStyleFile}[1]{%
  \renewcommand*\glstfilebase{#1}%
  \ifglxindy
    \def\istfilename{\glstfilebase.xdy}
  \else
    \def\istfilename{\glstfilebase.ist}
  \fi
}
\@onlypremakeg\setStyleFile
\ifglxindy
  \def\istfilename{\glstfilebase.xdy}
\else
  \def\istfilename{\glstfilebase.ist}
\fi
\newcommand*\glstfilebase{\jobname}
\newcommand*\@istfilename}[1]{%
\newcommand*\glscpositor}{.}
\newcommand*\glstSetCompositor}[1]{%
  \renewcommand*\glscpositor{#1}}
\@onlypremakeg\glstSetCompositor
\newcommand*\@glstAlphacompositor{\glscpositor}
\ifglxindy
  \newcommand*\glstSetAlphaCompositor[1]{%
    \renewcommand*\@glstAlphacompositor{#1}}
\else
  \newcommand*\glstSetAlphaCompositor[1]{%
    \glstnoxywarning\glstSetAlphaCompositor}
\fi
\@onlypremakeg\glstSetAlphaCompositor
\newcommand*\glstsuffixF{}
\newcommand*\glstSetSuffixF}[1]{%
  \renewcommand*\glstsuffixF{#1}}
\@onlypremakeg\glstSetSuffixF
\newcommand*\glstsuffixFF{}
\newcommand*\glstSetSuffixFF}[1]{%
  \renewcommand*\glstsuffixFF{#1}}
}
\ifcsundef{hyperlink}%
{%
  \newcommand*\glstnumberformat}[1]{#1}%
}%
{%
  \newcommand*\glstnumberformat}[1]{\glsthypernumber{#1}}%
}

```

```

\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*{\glossarypreamble}{%
  \csuse{@glossarypreamble@\currentglossary}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaultttype]{%
  \ifglossaryexists*{#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*{\glossarypostamble}{}
\newcommand*{\glossarysection}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@p@glossarysection{#1}{#2}%
  }%
  \gls@glossarymark{\glossarytoctitle}%
}
\ifcsundef{glossarymark}%
{%
  \newcommand{\gls@glossarymark}[1]{\glossarymark{#1}}
}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand{\gls@glossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUHead{#1}}{\memUHead{#1}}%
      \else
        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand{\gls@glossarymark}[1]{%
      \ifglsucmark
        \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
      \else
        \@mkboth{#1}{#1}%
      \fi
    }
  }
}
\providecommand{\glossarymark}[1]{%
  \ifglsucmark
    \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%

```

```

\else
  \@mkboth{#1}{#1}%
\fi
}
\newcommand*\setglossarysection[1]{%
\setkeys{glossaries.sty}{section=#1}}
\newcommand*\@glossarysection[2]{%
\ifdefempty\@glossarysecstar
{%
  \csname\@glossarysec\endcsname[#1]{#2}%
}%
{%
  \csname\@glossarysec\endcsname*{#2}%
  \gls@toc{#1}{\@glossarysec}%
}%
\@glossaryseclabel
}
\newcommand*\@pglossarysection[2]{%
\gls@clearpage
\phantomsection
\ifdefempty\@glossarysecstar
{%
  \csname\@glossarysec\endcsname{#2}%
}%
{%
  \gls@toc{#1}{\@glossarysec}%
  \csname\@glossarysec\endcsname*{#2}%
}%
\@glossaryseclabel
}
\newcommand*\gls@doclearpage{%
\ifthenelse{\equal{\@glossarysec}{chapter}}{%
{%
  \ifcsundef{cleardoublepage}%
  {%
    \clearpage
  }%
  {%
    \ifcsdef{if@openright}%
    {%
      \if@openright
        \cleardoublepage
      \else
        \clearpage
      \fi
    }%
  }%
  \cleardoublepage
}%
}

```

```

}%
{}%
}
\newcommand*\glsclearpage{\gls@doclearpage}
\newcommand*\@gls@toc}[2]{%
  \ifglstoc
    \ifglsnumberline
      \addcontentsline{toc}{#2}{\protect\numberline{#1}}%
    \else
      \addcontentsline{toc}{#2}{#1}%
    \fi
  \fi
}
\newcommand*\glsnoxindywarning}[1]{%
  \GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
}
\newcommand*\glsnomakeindexwarning}[1]{%
  \GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%
}
\ifglsxindy
  \edef\xdyattributes{\string"default\string"}%
\fi
\ifglsxindy
  \edef\xdyattributelist{}%
\fi
\ifglsxindy
  \def\xdylocref{}
\fi
\newcommand*\@gls@ifinlist}[4]{%
  \def\@do@ifinlist##1,#1,##2\end@doifinlist{%
    \def\@gls@listsuffix{##2}%
    \ifx\@gls@listsuffix\@empty
      #4%
    \else
      #3%
    \fi
  }%
  \@do@ifinlist,#2,#1,\end@doifinlist
}
\ifglsxindy
  \newcommand*\@xdycounters{\glscounter}
  \newcommand*\GlsAddXdyCounters[1]{%
    \@for\@gls@ctr:=#1\do{%
      \edef\@do@addcounter{%
        \noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
        {%
          \noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
            \noexpand\@gls@ctr}%
        }%
      }%
    }%
  }%

```

```

        \do@addcounter
    }
}
\@onlypremakeg\GlsAddXdyCounters
\else
\newcommand*\GlsAddXdyCounters[1]{%
\glsnoxywarning\GlsAddXdyAttribute
}
\fi
\newcommand*\@disabled@glssaddxdycounters{%
\PackageError{glossaries}{\string\GlsAddXdyCounters\space
can't be used after \string\GlsAddXdyAttribute}{Move all
occurrences of \string\GlsAddXdyCounters\space before the first
instance of \string\GlsAddXdyAttribute}%
}
\ifglxsindy
\newcommand*\@glssaddxyattribute[2]{%
\edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string" ^^J
\string"#2#1\string"}%
\expandafter\toks@\expandafter{\@xdylocref}%
\edef\@xdylocref{\the\toks@ ^^J%
(markup-locref
:open \string"\glstildechar n%
\expandafter\string\csname glsX#2X#1\endcsname
\string" ^^J
:close \string"\string" ^^J
:attr \string"#2#1\string")}%
\expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
\setentrycounter[##1]{#2}\csname #1\endcsname{##2}%
}%
}
\newcommand*\GlsAddXdyAttribute[1]{%
\ifx\@xdyattributelist\@empty
\edef\@xdyattributelist{#1}%
\else
\edef\@xdyattributelist{\@xdyattributelist,#1}%
\fi
\@for\@this@counter:=\@xdycounters\do{%
\protected@edef\gls@do@addxyattribute{%
\noexpand\@glssaddxyattribute{#1}{\@this@counter}%
}
\gls@do@addxyattribute
}%
\let\GlsAddXdyCounters\@disabled@glssaddxdycounters
}
\@onlypremakeg\GlsAddXdyAttribute
\else
\newcommand*\GlsAddXdyAttribute[1]{%
\glsnoxywarning\GlsAddXdyAttribute}
\fi

```

```

\ifglxindy
\newcommand*{\@gls@addpredefinedattributes}{%
  \GlsAddXdyAttribute{glsnumberformat}
  \GlsAddXdyAttribute{textrm}
  \GlsAddXdyAttribute{textsf}
  \GlsAddXdyAttribute{texttt}
  \GlsAddXdyAttribute{textbf}
  \GlsAddXdyAttribute{textmd}
  \GlsAddXdyAttribute{textit}
  \GlsAddXdyAttribute{textup}
  \GlsAddXdyAttribute{textsl}
  \GlsAddXdyAttribute{textsc}
  \GlsAddXdyAttribute{emph}
  \GlsAddXdyAttribute{glsnumber}
  \GlsAddXdyAttribute{hyperm}
  \GlsAddXdyAttribute{hypersf}
  \GlsAddXdyAttribute{hypertt}
  \GlsAddXdyAttribute{hyperbf}
  \GlsAddXdyAttribute{hypermd}
  \GlsAddXdyAttribute{hyperit}
  \GlsAddXdyAttribute{hyperup}
  \GlsAddXdyAttribute{hypersl}
  \GlsAddXdyAttribute{hypersc}
  \GlsAddXdyAttribute{hyperemph}
  \GlsAddXdyAttribute{glsignore}
}
\else
  \let\@gls@addpredefinedattributes\relax
\fi
\def\@xdyuseralphabets{}
\ifglxindy
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \edef\@xdyuseralphabets{%
      \@xdyuseralphabets ^^J
      (define-alphabet "#1" (#2))}
  \else
    \newcommand*{\GlsAddXdyAlphabet}[2]{%
      \glsnoxywarning\GlsAddXdyAlphabet}
  \fi
\ifglxindy
  \newcommand*{\@gls@xdy@locationlist}{%
    roman-page-numbers,%
    Roman-page-numbers,%
    arabic-page-numbers,%
    alpha-page-numbers,%
    Alpha-page-numbers,%
    Appendix-page-numbers,%
    arabic-section-numbers%
  }
\protected@edef\@gls@roman{\@roman{0}\string"

```

```

        \string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@gls@roman
\edef\@tmp{\string" \string"roman-numbers-lowercase\string"
:sep \string}}%
\@onelevel@sanitize\@tmp
\ifx\@tmp\@gls@roman
\expandafter
\edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
\string"roman-numbers-lowercase\string"%
}%
\else
\expandafter
\edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{
:sep \string"\@gls@roman\string"%
}%
\fi
\expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
\string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
\string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
\string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
\string"ALPHA\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
\string"ALPHA\string"
:sep \string"\@gls@Alphacompositor\string"
\string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
\string"arabic-numbers\string"
:sep \string"\glscompositor\string"
\string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglxindy
\newcommand\@xdycrossrefhook{}
\fi
\ifglxindy
\newcommand*\GlsAddXdyLocation}[3][[]]{%
\def\@gls@tmp{#1}%
\ifx\@gls@tmp\@empty
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%

```

```

        (define-location-class \string"#2\string"^^J\space\space
        \space(:sep \string"{}\glsoopenbrace\string" #3
        :sep \string"\glsclosebrace\string"))
    }%
\else
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep "\glsoopenbrace"
#1
:sep "\glsclosebrace\glsoopenbrace" #3
:sep "\glsclosebrace"))
}%
\fi
\edef\@xdyuserlocationnames{%
\@xdyuserlocationnames^^J\space\space\space
\string"#2\string"}%
}
\@onlypremakeg\GlsAddXdyLocation
\else
\newcommand*\GlsAddXdyLocation[2]{%
\glsnoxindywarning\GlsAddXdyLocation}
\fi
\ifglxindy
\def\@xdylocationclassorder{^^J\space\space\space
\string"roman-page-numbers\string"^^J\space\space\space
\string"arabic-page-numbers\string"^^J\space\space\space
\string"arabic-section-numbers\string"^^J\space\space\space
\string"alpha-page-numbers\string"^^J\space\space\space
\string"Roman-page-numbers\string"^^J\space\space\space
\string"Alpha-page-numbers\string"^^J\space\space\space
\string"Appendix-page-numbers\string"
\@xdyuserlocationnames^^J\space\space\space
\string"see\string"
}
\fi
\ifglxindy
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\def\@xdylocationclassorder{#1}}
\else
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\glsnoxindywarning\GlsSetXdyLocationClassOrder}
\fi
\fi
\ifglxindy
\def\@xdysortrules{}
\fi
\ifglxindy
\newcommand*\GlsAddSortRule[2]{%
\expandafter\toks@\expandafter{\@xdysortrules}%
\protected@edef\@xdysortrules{\the\toks@ ^^J

```



```

        (sort-rule \string"#1\string" \string"#2\string"))}%
    }
\else
  \newcommand*\GlsAddSortRule[2]{%
    \glsnoxywarning\GlsAddSortRule}
\fi
\ifglxindy
  \def\@xdyrequiredstyles{tex}
\fi
\ifglxindy
  \newcommand*\GlsAddXdyStyle[1]{%
    \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
  \newcommand*\GlsAddXdyStyle[1]{%
    \glsnoxywarning\GlsAddXdyStyle}
\fi
\ifglxindy
  \newcommand*\GlsSetXdyStyles[1]{%
    \edef\@xdyrequiredstyles{#1}}
\else
  \newcommand*\GlsSetXdyStyles[1]{%
    \glsnoxywarning\GlsSetXdyStyles}
\fi
\newcommand*\findrootlanguage{}
\def\@xdylanguage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
    \ifglossaryexists{#1}{%
      \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
    }{%
      \PackageError{glossaries}{Can't set language type for
        glossary type '#1' --- no such glossary}{%
        You have specified a glossary type that doesn't exist}}
\else
  \newcommand*\GlsSetXdyLanguage[2][{}]{%
    \glsnoxywarning\GlsSetXdyLanguage}
\fi
\def\@gls@codepage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyCodePage[1]{%
    \renewcommand*\@gls@codepage{#1}%
  }
  \AtBeginDocument{%
    \ifx\@gls@codepage\@empty
      \ifpackageloaded{fontspec}{\def\@gls@codepage{utf8}}{}%
    \fi
  }
\else
  \newcommand*\GlsSetXdyCodePage[1]{%
    \glsnoxywarning\GlsSetXdyCodePage}

```

```

\fi
\ifglxsindy
  \ifglxsxindy@glslnumbers
    \def\@xdylettergroups{(define-letter-group
      \string"glslnumbers\string"^^J\space\space\space
      :prefixes (\string"0\string" \string"1\string"
        \string"2\string" \string"3\string" \string"4\string"
        \string"5\string" \string"6\string" \string"7\string"
        \string"8\string" \string"9\string")^^J\space\space\space
      \@xdynumbergrouporder)}
    \else
      \def\@xdylettergroups{}
    \fi
  \fi
\fi
\newcommand*\GlsAddLetterGroup[2]{%
  \expandafter\toks@\expandafter{\@xdylettergroups}%
  \protected@edef\@xdylettergroups{\the\toks@^^J%
    (define-letter-group \string"#1\string"^^J\space\space\space#2)}%
  }%
\newcommand*\forallglossaries}[3][\@glo@types]{%
  \@for#2:=#1\do{\ifx#2\empty\else#3\fi}%
}

\newcommand*\forallacronyms}[2]{%
  \@for#1:=\@glslacronymlists\do{\ifx#1\empty\else#2\fi}%
}
\newcommand*\forallglsentries}[3][\@glsldefaulttype]{%
  \edef\@glo@list{\csname glolist@#1\endcsname}%
  \@for#2:=\@glo@list\do
  {%
    \ifdefempty{#2}{#3}%
  }%
}
\newcommand*\forallglsentries}[3][\@glo@types]{%
  \expandafter\forallglossaries\expandafter[#1]{\@this@glo@}%
  {%
    \forallglsentries[\@this@glo@]{#2}{#3}%
  }%
}
\newcommand{\ifglossaryexists}{%
  \@ifstar\s@ifglossaryexists\@ifglossaryexists
}
\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{@glo@type@#1@out}{#3}{#2}%
}
\newcommand{\s@ifglossaryexists}[3]{%
  \ifcsundef{glolist@#1}{#3}{#2}%
}
\newcommand*\glsdetoklabel}[1]{#1}
\newcommand{\ifglsentryexists}[3]{%

```

```

\ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}
\newcommand*{\ifglsused}[3]{%
\ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}
\newcommand{\glsdoifexists}[2]{%
\ifglsentryexists{#1}{#2}{%
\PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
has not been defined}{You need to define a glossary entry before you
can use it.}}%
}
\newcommand{\glsdoifnoexists}[2]{%
\ifglsentryexists{#1}{%
\PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
been defined}{}}#2}%
}
\newcommand{\glsdoifexistsorwarn}[2]{%
\ifglsentryexists{#1}{#2}{%
\GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
has not been defined}%
}%
}
\newcommand{\glsdoifexistsordo}[3]{%
\ifglsentryexists{#1}{#2}{%
\PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
has not been defined}{You need to define a glossary entry before you
can use it.}}%
#3%
}%
}
\newcommand{\doifglossarynoexistsordo}[3]{%
\ifglossaryexists*{#1}%
{%
\PackageError{glossaries}{Glossary type ‘#1’ already exists}{}%
#3%
}%
#2}%
}
\newrobustcmd{\ifglshaschildren}[3]{%
\glsdoifexists{#1}%
{%
\def\do@glshaschildren{#3}%
\edef@gl@thislabel{\glsdetoklabel{#1}}%
\expandafter\forglseries\expandafter
[\csname glo@\@gl@thislabel @type\endcsname]
{\glo@label}%
{%
\letcs\glo@parent{glo@\glo@label @parent}%
\ifdefequal\@gl@thislabel\glo@parent
{%

```

```

        \def\do@glshaschildren{#2}%
        \@endfortrue
    }%
    {}%
    }%
    \do@glshaschildren
}
}
\newcommand{\ifglshasparent}[3]{%
    \glsdoifexists{#1}%
    {%
        \ifcsempy{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
    }%
}
\newcommand*{\ifglshasdesc}[3]{%
    \ifcsempy{glo@\glsdetoklabel{#1}@desc}%
    {#3}%
    {#2}%
}
\newcommand*{\ifglshasdescsuppressed}[3]{%
    \ifcsequal{glo@\glsdetoklabel{#1}@desc}{@no@post@desc}%
    {#2}%
    {#3}%
}
}
\newrobustcmd*{\ifglshassymbol}[3]{%
    \letcs{\@glo@symbol}{glo@\glsdetoklabel{#1}@symbol}%
    \ifdefempty\@glo@symbol
    {#3}%
    {%
        \ifdefequal\@glo@symbol\@gls@default@value
        {#3}%
        {#2}%
    }%
}
}
\newrobustcmd*{\ifglshaslong}[3]{%
    \letcs{\@glo@long}{glo@\glsdetoklabel{#1}@long}%
    \ifdefempty\@glo@long
    {#3}%
    {%
        \ifdefequal\@glo@long\@gls@default@value
        {#3}%
        {#2}%
    }%
}
}
\newrobustcmd*{\ifglshasshort}[3]{%
    \letcs{\@glo@short}{glo@\glsdetoklabel{#1}@short}%
    \ifdefempty\@glo@short
    {#3}%
    {%
        \ifdefequal\@glo@short\@gls@default@value

```

```

    {#3}%
    {#2}%
  }%
}
\newrobustcmd*{\ifglshasfield}[4]{%
  \glsdoifexists{#2}%
  {%
    \letcs{\@glo@thisvalue}{glo@glstetoklabel{#2}@#1}%
    \ifdef\@glo@thisvalue
    {%
      \ifdefempty\@glo@thisvalue
      {%
        #4%
      }%
    }%
    {%
      \ifdefequal\@glo@thisvalue\@gls@default@value
      {%
        #4%
      }%
    }%
    \let\glscurrentfieldvalue\@glo@thisvalue
    #3%
  }%
}%
}%
{%
  \@gls@fetchfield{\@gls@thisfield}{#1}%
  \ifdef\@gls@thisfield
  {%
    \letcs{\@glo@thisvalue}{glo@glstetoklabel{#2}@#1}%
    \ifdefempty\@glo@thisvalue
    {%
      #4%
    }%
  }%
  {%
    \ifdefequal\@glo@thisvalue\@gls@default@value
    {%
      #4%
    }%
  }%
  \let\glscurrentfieldvalue\@glo@thisvalue
  #3%
}%
}%
}%
\GlossariesWarning{Unknown entry field '#1'}%
#4%
}%
}%

```

```

}%
}
\newcommand*\glscurrentfieldvalue{}
\newcommand*\@glo@types{,}
\newcommand*\@gls@provide@newglossary{%
  \protected@write\@auxout{}\string\providecommand\string\@newglossary[4]{}%
  \let\@gls@provide@newglossary\relax
}
\newcommand*\defglsentryfmt}[2][\glsdefaulttype]{%
  \csgdef{gls@#1@entryfmt}{#2}%
}
\newcommand*\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*\@gls@forbidtext}[1]{%
  \ifboolexpr{test {\ifdefstring{#1}{tex}}
    or test {\ifdefstring{#1}{TEX}}}
  {%
    \def#1{nottex}%
    \PackageError{glossaries}%
      {Forbidden '.tex' extension replaced with '.nottex'}%
      {I'm sorry, I can't allow you to do something so reckless.\MessageBreak
        Don't use '.tex' as an extension for a temporary file.}%
  }%
  {%
  }%
}
\newcommand*\@gls@gobbleopt{\new@ifnextchar[{\@gls@gobbleopt}{}]}
\def\@gls@gobbleopt[#1]{
\newcommand*\@newglossary{\@ifstar\s@newglossary\ns@newglossary}
\newcommand*\@s@newglossary}[2]{%
  \ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
}
\newcommand*\@ns@newglossary}[5][glg]{%
  \doifglossarynoexistsordo{#2}%
  {%
    \ifundef\glsdefaulttype
    {%
      \gdef\glsdefaulttype{#2}%
    }-}%
  \toks@{#2}\edef\@glo@types{\@glo@types\the\toks@,}%
  \expandafter\gdef\csname glolist@#2\endcsname{,}%
  \expandafter\edef\csname @glo@type@#2@log\endcsname{#1}%
  \expandafter\edef\csname @glo@type@#2@in\endcsname{#3}%
  \expandafter\edef\csname @glo@type@#2@out\endcsname{#4}%
  \expandafter\@gls@forbidtext\csname @glo@type@#2@log\endcsname
  \expandafter\@gls@forbidtext\csname @glo@type@#2@in\endcsname
  \expandafter\@gls@forbidtext\csname @glo@type@#2@out\endcsname
  \expandafter\def\csname @glo@type@#2@title\endcsname{#5}%
  \@gls@provide@newglossary
  \protected@write\@auxout{}\string\@newglossary{#2}{#1}{#3}{#4}%
  \ifcsundef{gls@#2@entryfmt}%

```

```

{%
  \defglsentryfmt[#2]{\glsentryfmt}%
}%
{}%
\@gls@defsortcount{#2}%
\@ifnextchar[{\@gls@setcounter{#2}}%
  {\@gls@setcounter{#2}{\glscounter}}%
}%
{%
  \gls@gobbleopt
}%
}
\newcommand*\altnewglossary}[3]{%
  \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
}
\@onlypreamble{\newglossary}
\@onlypremakeg\newglossary
\newcommand*\@newglossary}[4]{}
\def\@gls@setcounter#1[#2]{%
  \expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
  \ifglsxindy
    \GlsAddXdyCounters{#2}%
  \fi
}
\newcommand*\@gls@getcounter}[1]{%
  \csname @glotype@#1@counter\endcsname
}
\glsdefmain
\@gls@do@acronymsdef
\@gls@do@symbolsdef
\@gls@do@numbersdef
\@gls@do@indexdef
\newcommand*\newignoredglossary}[1]{%
  \ifdefempty\@ignored@glossaries
  {%
    \edef\@ignored@glossaries{#1}%
  }%
  {%
    \eappto\@ignored@glossaries{,#1}%
  }%
  \csgdef{glolist@#1}{,}%
  \ifcsundef{gls@#1@entryfmt}%
  {%
    \defglsentryfmt[#1]{\glsentryfmt}%
  }%
  {}%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*\@gls@nohyperlist}{#1}%
  }%

```

```

    {%
      \eappto\@gls@nohyperlist{,#1}%
    }%
  }
  \newcommand*{\@ignored@glossaries}{}
  \newcommand*{\ifignoredglossary}[3]{%
    \edef\@gls@igtype{#1}%
    \expandafter\DTLifinlist\expandafter
      {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
  }
  \define@key{glossentry}{name}{%
    \def\@glo@name{#1}%
  }
  \define@key{glossentry}{description}{%
    \def\@glo@desc{#1}%
  }
  \define@key{glossentry}{descriptionplural}{%
    \def\@glo@descplural{#1}%
  }
  \define@key{glossentry}{sort}{%
    \def\@glo@sort{#1}}
  \define@key{glossentry}{text}{%
    \def\@glo@text{#1}%
  }
  \define@key{glossentry}{plural}{%
    \def\@glo@plural{#1}%
  }
  \define@key{glossentry}{first}{%
    \def\@glo@first{#1}%
  }
  \define@key{glossentry}{firstplural}{%
    \def\@glo@firstplural{#1}%
  }
  \newcommand*{\@gls@default@value}{\relax}
  \define@key{glossentry}{symbol}{%
    \def\@glo@symbol{#1}%
  }
  \define@key{glossentry}{symbolplural}{%
    \def\@glo@symbolplural{#1}%
  }
  \define@key{glossentry}{type}{%
    \def\@glo@type{#1}}
  \define@key{glossentry}{counter}{%
    \ifcsundef{c@#1}%
      {%
        \PackageError{glossaries}%
          {There is no counter called ‘#1’}%
          {%
            The counter key should have the name of a valid counter
            as its value%
          }
      }
    }

```



```

    }%
  }%
  {%
    \def\@glo@counter{#1}%
  }%
}
\define@key{glossentry}{see}{%
  \gls@set@xr@key{see}{\@glo@see}{#1}%
}
\newcommand*\@gls@set@xr@key}[3]{%
  \renewcommand*\@gls@xr@key{#1}%
  \gls@checkseeallowed
  \def#2{#3}%
  \@glo@seeautonumberlist
}
\newcommand*\@gls@xr@key{see}
\newcommand*\@gls@checkseeallowed{%
  \@gls@see@noindex
}
\newcommand*\@gls@checkseeallowed@preambleonly{%
  \GlossariesWarning{glossaries}%
  {'\@gls@xr@key' key doesn't have any effect when used in the document
  environment. Move the definition to the preamble
  after \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@key{glossentry}{parent}{%
  \def\@glo@parent{#1}}
\define@choicekey{glossentry}{nonumberlist}{%
  [\@gls@nonumberlist@val\@gls@nonumberlist@nr]{true,false}[true]}%
}%
  \ifcase\@gls@nonumberlist@nr\relax
    \def\@glo@prefix{\@gls@nonextpages}%
    \@gls@savenonumberlist{true}%
  \else
    \def\@glo@prefix{\@gls@nextpages}%
    \@gls@savenonumberlist{false}%
  \fi
}
\newcommand*\@gls@savenonumberlist}[1]{}
\newcommand*\@gls@initnonumberlist[1]{}
\newcommand*\@gls@storenonumberlist[1]{}
\newcommand*\@gls@enablesavenonumberlist[1]{%
  \renewcommand*\@gls@initnonumberlist[1]{}
  \undef\@glo@nonumberlist
}%
\renewcommand*\@gls@savenonumberlist}[1]{%
  \def\@glo@nonumberlist{##1}%
}%
\renewcommand*\@gls@storenonumberlist}[1]{%

```

```

\ifdef\@glo@nonumberlist
{%
  \cslet{glo\@glsdetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
}%
{}%
}%
\appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}
\define@key{glossentry}{user1}{%
  \def\@glo@useri{#1}%
}
\define@key{glossentry}{user2}{%
  \def\@glo@userii{#1}%
}
\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#1}%
}
\define@key{glossentry}{short}{%
  \def\@glo@short{#1}%
}
\define@key{glossentry}{shortplural}{%
  \def\@glo@shortpl{#1}%
}
\define@key{glossentry}{long}{%
  \def\@glo@long{#1}%
}
\define@key{glossentry}{longplural}{%
  \def\@glo@longpl{#1}%
}
\newcommand*\@glsnoname}{%
  \PackageError{glossaries}{name key required in
  \string\newglossaryentry\space for entry '\@glo@label'}{You
  haven't specified the entry name}}
\newcommand*\@glsnodesc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '\@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}

```

```

}%
}%
\newcommand*{\@glsdefaultplural}{%
\newcommand*{\@gls@missingnumberlist}[1]{%
??%
\ifgls@savenumberlist
\GlossariesWarning{Missing number list for entry '#1'.
Maybe makeglossaries + rerun required}%
\else
\PackageError{glossaries}%
{Package option 'savenumberlist=true' required}%
{%
You must use the 'savenumberlist' package option
to reference location lists.%
}%
\fi
}
\newcommand*{\@glsdefaultsort}{\@glo@name}
\newcount\gls@level
\newcommand{\@gls@noexpand@field}[3]{%
\expandafter\global\expandafter
\let\csname glo@#1@#2\endcsname#3%
}
\newcommand{\@gls@noexpand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\csuse{gls@assign@#3@field}{#2}{#4}%
}%
}%
}
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\@gls@noexpand@field{#2}{#3}{\@gls@value}%
}%
{%
\@gls@noexpand@field{#2}{#3}{#4}%
}%
}%
}
\newcommand{\@gls@expand@field}[3]{%
\expandafter
\protected@xdef\csname glo@#1@#2\endcsname{#3}%
}

```

```

\newcommand{\@gls@expand@fields}[4]{%
  \ifcsdef{gls@assign@#3@field}
  {%
    \ifdequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
    }%
    {%
      \expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
      {%
        \@gls@expand@field{#2}{#3}{#4}%
      }%
      {%
        \csuse{gls@assign@#3@field}{#2}{#4}%
      }%
    }%
  }%
}
\ifdequal{#4}{\@gls@default@value}%
{%
  \@gls@expand@field{#2}{#3}{#1}%
}%
{%
  \@gls@expand@field{#2}{#3}{#4}%
}%
}%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
  \def\@gls@tmp{#1}%
  \ifdequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*\gls@expandfields{%
  \let\gls@assign@field\@gls@expand@fields
}
\newcommand*\gls@noexpandfields{%
  \let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
  \glsdoifnoexists{#1}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\newcommand*\gls@defdocnewglossaryentry{%
  \let\gls@checkseeallowed\gls@checkseeallowed@preambleonly
  \let\newglossaryentry\new@glossaryentry
}

```

```

\newrobustcmd{\provideglossaryentry}[2]{%
  \ifglentryexists{#1}%
  {}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\@onlypreamble{\provideglossaryentry}
\newrobustcmd{\new@glossaryentry}[2]{%
  \ifundef\@gls@deffile
  {%
    \global\newwrite\@gls@deffile
    \immediate\openout\@gls@deffile=\jobname.glsdefs
  }%
  {}%
  \ifglentryexists{#1}{}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
  \@gls@writedef{#1}%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*\@gls@begindocdefs{%
  \@gls@enablesavenonumberlist
  \edef\@gls@restoreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
  \makeatletter
  \InputIfFileExists{\jobname.glsdefs}{%}{%
  \@gls@restoreat
  \undef\@gls@restoreat
  \gls@defdocnewglossaryentry
}
\newcommand*\@gls@writedef}[1]{%
  \immediate\write\@gls@deffile
  {%
    \string\ifglentryexists{#1}{}\glspercentchar^^J%
    \expandafter\@gobble\string{\glspercentchar^^J%
      \string\gls@defglossaryentry{\glsdetoklabel{#1}}\glspercentchar^^J%
      \expandafter\@gobble\string{\glspercentchar%
}
}
\@for\@gls@map:=\@gls@keymap\do
{%
  \letcs\glo@value{glo@\glsdetoklabel{#1}}\expandafter\@secondoftwo\@gls@map}%
  \ifdef\glo@value
  {%
    \@onelevel@sanitize\glo@value
    \immediate\write\@gls@deffile
    {%
      \expandafter\@firstoftwo\@gls@map
      =\expandafter\@gobble\string{\glo@value\expandafter\@gobble\string\},%
}
}
}

```

```

        \glspercentchar
    }%
    }%
    {}%
}%
\gls.writedefhook
\immediate\write\@gls@deffile
{%
    \glspercentchar^^J%
    \expandafter\@gobble\string\}\glspercentchar^^J%
    \expandafter\@gobble\string\}\glspercentchar%
}%
}
\newcommand*\@gls@keymap{%
    {name}{name},%
    {sort}{sortvalue},% unescaped sort value
    {type}{type},%
    {first}{first},%
    {firstplural}{firstpl},%
    {text}{text},%
    {plural}{plural},%
    {description}{desc},%
    {descriptionplural}{descplural},%
    {symbol}{symbol},%
    {symbolplural}{symbolplural},%
    {user1}{useri},%
    {user2}{userii},%
    {user3}{useriii},%
    {user4}{useriv},%
    {user5}{userv},%
    {user6}{uservi},%
    {long}{long},%
    {longplural}{longpl},%
    {short}{short},%
    {shortplural}{shortpl},%
    {counter}{counter},%
    {parent}{parent}%
}
\newcommand*\@gls@fetchfield}[2]{%
    \edef\@gls@thisval{#2}%
    \for\@gls@map:=\@gls@keymap\do{%
        \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
        \ifdefequal{\@this@key}{\@gls@thisval}%
        {%
            \edef#1{\expandafter\@secondoftwo\@gls@map}%
            \@endfortrue
        }%
    }%
}%
}

```

```

\newcommand*\glsaddstoragekey{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
\newcommand*\@sglsaddstoragekey[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
  {}%
  \@glsaddstoragekey{#1}%
}
\newcommand*\@glsaddstoragekey[3]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{, {#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
    \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
  }%
  {}%
  \PackageError{glossaries}{Key ‘#1’ already exists}{}%
  }%
}
\newcommand*\glsaddkey{\@ifstar\@sglsaddkey\@glsaddkey}
\newcommand*\@sglsaddkey[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
  {}%
  \@glsaddkey{#1}%
}
\newcommand*\@glsaddkey[7]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{, {#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
  }%
}

```

```

\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
\newcommand*{#4}[1]{\@Gls@entry@field{##1}{#1}}%
\ifcsdef{@gls@user@#1@}%
{
  \PackageError{glossaries}%
  {Can't define '\string#5' as helper command
   '\expandafter\string\csname @gls@user@#1@\endcsname' already exists}%
  {}%
}%
{
  \expandafter\newcommand\expandafter*\expandafter
  {\csname @gls@user@#1@\endcsname}[2][]{%
   \new@ifnextchar[%
    {\csuse{@gls@user@#1@}{##1}{##2}}%
    {\csuse{@gls@user@#1@}{##1}{##2}[[]]}%
  \csdef{@gls@user@#1@}##1##2[##3]{%
   \@gls@field@link{##1}{##2}{#3{##2}##3}%
  }%
  \newrobustcmd*{#5}{%
   \expandafter\@gls@hyp@opt\csname @gls@user@#1@\endcsname}%
  }%
\ifcsdef{@Gls@user@#1@}%
{
  \PackageError{glossaries}%
  {Can't define '\string#6' as helper command
   '\expandafter\string\csname @Gls@user@#1@\endcsname' already exists}%
  {}%
}%
{
  \expandafter\newcommand\expandafter*\expandafter
  {\csname @Gls@user@#1@\endcsname}[2][]{%
   \new@ifnextchar[%
    {\csuse{@Gls@user@#1@}{##1}{##2}}%
    {\csuse{@Gls@user@#1@}{##1}{##2}[[]]}%
  \csdef{@Gls@user@#1@}##1##2[##3]{%
   \@gls@field@link{##1}{##2}{#4{##2}##3}%
  }%
  \newrobustcmd*{#6}{%
   \expandafter\@gls@hyp@opt\csname @Gls@user@#1@\endcsname}%
  }%
\ifcsdef{@GLS@user@#1@}%
{
  \PackageError{glossaries}%
  {Can't define '\string#7' as helper command
   '\expandafter\string\csname @GLS@user@#1@\endcsname' already exists}%
  {}%
}%
{
  \expandafter\newcommand\expandafter*\expandafter
  {\csname @GLS@user@#1@\endcsname}[2][]{%

```



```

        \new@ifnextchar [%
            {\csuse{@GLS@user@#1@}{##1}{##2}}%
            {\csuse{@GLS@user@#1@}{##1}{##2}[]}%
        \csdef{@GLS@user@#1@}##1##2[##3]{%
            \@gls@field@link{##1}{##2}{\mfirstucMakeUppercase{#3{##2}##3}}%
        }%
        \newrobustcmd*{#7}{%
            \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
        }%
    }%
}
{
    \PackageError{glossaries}{Key ‘#1’ already exists}{%
}
}
\newcommand{\glsfieldxddef}[3]{%
\glsdoifexists{#1}%
{
    \edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {
        \protected@csxdef{glo@\@glo@label @#2}{#3}%
    }%
    {
        \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{%
    }%
}
}
\newcommand{\glsfielddedef}[3]{%
\glsdoifexists{#1}%
{
    \edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {
        \protected@csedef{glo@\@glo@label @#2}{#3}%
    }%
    {
        \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{%
    }%
}
}
\newcommand{\glsfielddgdef}[3]{%
\glsdoifexists{#1}%
{
    \edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {
        \expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
    }%
    {
        \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{%
}
}

```

```

    }%
  }%
}
\newcommand{\glsfielddef}[3]{%
\glsdoifexists{#1}%
{%
  \edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand{\glsfieldfetch}[3]{%
\glsdoifexists{#1}%
{%
  \edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \letcs#3{glo@\@glo@label @#2}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand{\ifglsfieldeq}[5]{%
\glsdoifexists{#1}%
{%
  \edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand{\ifglsfielddefeq}[5]{%
\glsdoifexists{#1}%
{%
  \edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \expandafter\ifdefstrequal
      \csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
  }%
}

```

```

}%
{%
  \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
}%
}%
}
\newcommand{\ifglsfieldcseq}[5]{%
\glsdoifexists{#1}%
{%
  \edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \ifcsstrequal{glo@\@glo@label @#2}{#3}{#4}{#5}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand*{\gls.writedefhook}{}
\newcommand*{\gls@assign@desc}[1]{%
  \gls@assign@field{}{#1}{desc}{\@glo@desc}%
  \gls@assign@field{\@glo@desc}{#1}{descplural}{\@glo@descplural}%
}
\newcommand{\longnewglossaryentry}[3]{%
  \glsdoifnoexists{#1}%
  {%
    \bgroup
      \let\@org@newglossaryentryprehook\@newglossaryentryprehook
      \long\def\@newglossaryentryprehook{%
        \long\def\@glo@desc{#3\leavevmode\unskip\nopostdesc}%
        \@org@newglossaryentryprehook
      }%
      \renewcommand*{\gls@assign@desc}[1]{%
        \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
        \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@desc}%
      }
      \gls@defglossaryentry{#1}{#2}%
    \egroup
  }
}
\@onlypreamble{\longnewglossaryentry}
\newcommand{\longprovideglossaryentry}[3]{%
  \ifglsentryexists{#1}{}%
  {\longnewglossaryentry{#1}{#2}{#3}}%
}
\@onlypreamble{\longprovideglossaryentry}
\newcommand{\gls@defglossaryentry}[2]{%
  \let\GlsSetQuote\gls@nosetquote
  \edef\@glo@label{\glsdetoklabel{#1}}%

```

```

\let\glslabel\@glo@label
\let\@glo@name\@gls@name
\let\@glo@desc\@gls@desc
\let\@glo@descplural\@gls@descplural
\let\@glo@type\@gls@type
\let\@glo@symbol\@gls@symbol
\let\@glo@symbolplural\@gls@symbolplural
\let\@glo@text\@gls@text
\let\@glo@plural\@gls@plural
\let\@glo@first\@gls@first
\let\@glo@firstplural\@gls@firstplural
\let\@glo@sort\@gls@sort
\let\@glo@counter\@gls@counter
\def\@glo@see{ }%
\def\@glo@parent{ }%
\def\@glo@prefix{ }%
\@gls@initnonnumberlist
\def\@glo@useri{ }%
\def\@glo@userii{ }%
\def\@glo@useriii{ }%
\def\@glo@useriv{ }%
\def\@glo@userv{ }%
\def\@glo@uservi{ }%
\def\@glo@short{ }%
\def\@glo@shortpl{ }%
\def\@glo@long{ }%
\def\@glo@longpl{ }%
\@newglossaryentryprehook
\setkeys{glossentry}{#2}%
\ifundef\glsdefaulttype
{%
  \PackageError{glossaries}%
  {No default glossary type (have you used ‘nomain’ by mistake?)}%
  {If you use package option ‘nomain’ you must define
   a new glossary before you can define entries}%
}%
{}%
\gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
\edef\@glo@type{\glsentrytype{\@glo@label}}%
\ifcsundef{glo@list@\@glo@type}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘\@glo@type’ has not been defined}%
  {You need to define a new glossary type, before making entries
   in it}%
}%
{}%
\ifignoredglossary\@glo@type
{%
  \ifx\@glo@desc\@gls@desc

```

```

        \let\@glo@desc\@empty
    \fi
}%
{
}%
\protected@edef\@glo@list@\csname glo@list@\@glo@type\endcsname}%
\expandafter\xdef\csname glo@list@\@glo@type\endcsname{%
    \@glo@list@\@glo@label},}%
}%
\gls@level=0\relax
\ifx\@glo@parent\@empty
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
\else
    \ifdefequal\@glo@label\@glo@parent%
    {
        \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
        \def\@glo@parent{}%
        \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
    }%
    {
        \ifglsentryexists{\@glo@parent}%
        {
            \expandafter\xdef\csname glo@\@glo@label @parent\endcsname{%
                \@glo@parent}%
            \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
            \advance\gls@level by 1\relax
            \ifx\@glo@name\@glsnoname
                \expandafter\let\expandafter\@glo@name
                \csname glo@\@glo@parent @name\endcsname
            \ifx\@glo@plural\@gls@default@value
                \expandafter\let\expandafter\@glo@plural
                \csname glo@\@glo@parent @plural\endcsname
            \fi
        }
        \fi
    }%
    {
        \PackageError{glossaries}%
        {
            Invalid parent ‘\@glo@parent’
            for entry ‘\@glo@label’ - parent doesn’t exist%
        }%
        {
            Parent entries must be defined before their children%
        }%
        \def\@glo@parent{}%
        \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
    }%
}%
\fi
\expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%

```

```

\gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
\letcs@glo@sort{glo@\@glo@label @sortvalue}%
\gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @text\endcsname\glspluralsuffix}%
  {\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @text\endcsname}%
  {\@glo@label}{first}{\@glo@first}%
\ifx@glo@first@gls@default@value
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @plural\endcsname}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\else
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\fi
\ifcsundef{glo@type@\@glo@type @counter}%
{%
  \def@glo@defaultcounter{\glscounter}%
}%
\letcs@glo@defaultcounter{glo@type@\@glo@type @counter}%
}%
\gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
\gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
\gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
\gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
\gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
\gls@assign@field{}{\@glo@label}{userv}{\@glo@userv}%
\gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
\gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
\gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
\gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
\gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%
\ifx@glo@name@glsnoname
  \glsnoname
  \let@glo@name@gls@default@value
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%
\ifcsundef{glo@\@glo@label @numberlist}%
{%
  \csxdef{glo@\@glo@label @numberlist}{%
    \noexpand@gls@missingnumberlist{\@glo@label}}%
}%
\gls@storenonumberlist{\@glo@label}%
\def@glo@desc{\@glo@first}%
\ifx@glo@desc@glo@desc

```

```

\let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@glsnodesc
\@glsnodesc
\let\@glo@desc\@gls@default@value
\fi
\gls@assign@desc{\@glo@label}%
\@gls@defsort{\@glo@type}{\@glo@label}%
\def\@glo@@symbol{\@glo@text}%
\ifx\@glo@symbol\@glo@@symbol
\let\@glo@symbol\@glo@text
\fi
\gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
\expandafter
\gls@assign@field\expandafter
{ \csname glo@\@glo@label @symbol\endcsname }
{ \@glo@label } { symbolplural } { \@glo@symbolplural } %
\expandafter\xdef\csname glo@\@glo@label @flagfalse\endcsname{%
\noexpand\global
\noexpand\let\expandafter\noexpand
\csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
}%
\expandafter\xdef\csname glo@\@glo@label @flagtrue\endcsname{%
\noexpand\global
\noexpand\let\expandafter\noexpand
\csname ifglo@\@glo@label @flag\endcsname\noexpand\ifftrue
}%
\csname glo@\@glo@label @flagfalse\endcsname
\@glo@autosee
\ifignoredglossary\@glo@type
{%
\csdef{glo@\@glo@label @index}{}%
}
{%
\do@glo@storeentry{\@glo@label}%
}%
\@newglossaryentry@defcounters
\@newglossaryentry@posthook
}
\newcommand*{\@glo@autosee}{%
\ifdefvoid\@glo@see{%
{%
\protected@edef\@do@glssee{%
\noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
\noexpand\expandafter\noexpand\@glssee\noexpand\@glo@list{\@glo@label}}%
\do@glssee
}%
\@glo@autoseehook
}%
\newcommand*{\@glo@autoseehook}{%

```

```

\newcommand*{\@newglossaryentryprehook}{}
\newcommand*{\@newglossaryentryposthook}{}
\newcommand*{\@newglossaryentry@defcounters}{}
\newcommand*{\glsmoveentry}[2]{%
  \edef\@glo@thislabel{\glsdetoklabel{#1}}%
  \edef\@glo@type{\csname glo@\@glo@thislabel @type\endcsname}%
  \def\@glo@list{,%}
  \for@gl@entries[\@glo@type]{\@glo@label}%
  {%
    \if@defequal\@glo@thislabel\@glo@label
      {}{\@eappto\@glo@list{\@glo@label,}}%
    }%
  \cslet\@glo@list[\@glo@type]{\@glo@list}%
  \csdef{glo@\@glo@thislabel @type}{#2}%
}
\if@gl@xindy
  \newcommand*{\@glossaryentryfield}{\string\@glossentry}
\else
  \newcommand*{\@glossaryentryfield}{\string\@glossentry}
\fi
\if@gl@xindy
  \newcommand*{\@glossarysubentryfield}{%
    \string\@subglossentry}
\else
  \newcommand*{\@glossarysubentryfield}{%
    \string\@subglossentry}
\fi
\newcommand{\@glo@storeentry}[1]{%
  \edef\@glo@esclabel{#1}%
  \@gls@checkmkidxchars\@glo@esclabel
  \protected@edef\@glo@sort{\csname glo@#1@sort\endcsname}%
  \@gls@checkmkidxchars\@glo@sort
  \@gls@checkmkidxchars\@glo@prefix
  \edef\@glo@parent{\csname glo@#1@parent\endcsname}%
  \if@gl@xindy
    \ifx\@glo@parent\@empty
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        (\string\@glo@sort\string" %
          \string\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %
        }%
    \else
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \csname glo@\@glo@parent @index\endcsname
        (\string\@glo@sort\string" %
          \string\@glo@prefix\@glossarysubentryfield
            {\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
        }%
    \fi
  \else
    \ifx\@glo@parent\@empty

```



```

        \onelevel@sanitize\@glo@prefix
        \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
            \@glo@sort\@gls@actualchar\@glo@prefix
            \@glossaryentryfield{\@glo@esclabel}%
        }%
    \else
        \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
            \csname glo@\@glo@parent @index\endcsname\@gls@levelchar
            \@glo@sort\@gls@actualchar\@glo@prefix
            \@glossarysubentryfield
                {\csname glo@#1@level\endcsname}{\@glo@esclabel}%
            }%
        \fi
    \fi
}
\AtBeginDocument{%
    \ifpackageloaded{amsmath}%
    {\let\gls@ifnotmeasuring\@gls@ifnotmeasuring}%
    {}%
}
\newcommand*\@gls@ifnotmeasuring}[1]{%
    \ifmeasuring@
    \else
    #1%
    \fi
}
\newcommand*\gls@ifnotmeasuring[1]{#1}
\def\@gls@patchtabularx#1\hbox#2#3!!{%
    \def\TX@trial##1{#1\hbox{\let\glsunset\@gobble#2}#3}%
}
\newcommand*\glspatchtabularx{%
    \ifdef\TX@trial
    {%
        \expandafter\@gls@patchtabularx\TX@trial{##1}!!%
        \let\glspatchtabularx\relax
    }%
    {}%
}
\newcommand*\glsreset}[1]{%
    \gls@ifnotmeasuring
    {%
        \glsdoifexists{#1}%
        {%
            \@glsreset{#1}%
        }%
    }%
}
\newcommand*\glslocalreset}[1]{%
    \gls@ifnotmeasuring
    {%

```

```

\glsdoifexists{#1}%
{%
  \@glslocalreset{#1}%
}%
}%
}
\newcommand*\glsunset}[1]{%
\gls@ifnotmeasuring
{%
  \glsdoifexists{#1}%
  {%
    \@glsunset{#1}%
  }%
}%
}
\newcommand*\glslocalunset}[1]{%
\gls@ifnotmeasuring
{%
  \glsdoifexists{#1}%
  {%
    \@glslocalunset{#1}%
  }%
}%
}
\newcommand*\@glslocalunset{\@glslocalunset}
\newcommand*\@glslocalunset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
}
\newcommand*\@glsunset{\@glsunset}
\newcommand*\@glsunset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
}
\newcommand*\@glslocalreset{\@glslocalreset}
\newcommand*\@glslocalreset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
}
\newcommand*\@glsreset{\@glsreset}
\newcommand*\@glsreset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
}
\newcommand*\glsresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsreset{\@glsentry}%
  }%
}
\newcommand*\glslocalresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalreset{\@glsentry}%
  }%
}

```

```

}%
}
\newcommand*\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsunset{\@glsentry}%
  }%
}
\newcommand*\glslocalunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalunset{\@glsentry}%
  }%
}
\newcommand*\@@newglossaryentry@defcounters{%
  \csdef{glo@\@glo@label @currcount}{0}%
  \csdef{glo@\@glo@label @prevcount}{0}%
}
\newcommand*\glsenableentrycount{%
  \let\@newglossaryentry@defcounters\@@newglossaryentry@defcounters
  \renewcommand*\gls@defdocnewglossaryentry{%
    \renewcommand*newglossaryentry[2]{%
      \PackageError{glossaries}{\string\newglossaryentry\space
        may only be used in the preamble when entry counting has
        been activated}{If you use \string\glsenableentrycount\space
        you must place all entry definitions in the preamble not in
        the document environment}%
    }%
  }%
}
\newcommand*\glsentrycurrcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
  {0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*\glsentryprevcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
  {0}{\@gls@entry@field{##1}{prevcount}}%
}%
\renewcommand*\@glsunset}[1]{%
  \@glsunset{##1}%
  \@gls@increment@currcount{##1}%
}%
\renewcommand*\@glslocalunset}[1]{%
  \@glslocalunset{##1}%
  \@gls@local@increment@currcount{##1}%
}%
\renewcommand*\@glsreset}[1]{%
  \@glsreset{##1}%
  \csgdef{glo@\glsdetoklabel{##1}@currcount}{0}%
}%
\renewcommand*\@glslocalreset}[1]{%

```

```

    \@glslocalreset{##1}%
    \csdef{glo@glsdetoklabel{##1}@currcount}{0}%
  }%
\def\@cgl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cglformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@gls@{##1}{##2}[##3]%
  \fi
}%
\def\@cgl@spl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cgl@splformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@gls@{##1}{##2}[##3]%
  \fi
}%
\def\@cGls@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cGlsformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@Gls@{##1}{##2}[##3]%
  \fi
}%
\def\@cGls@spl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cGls@splformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@Gls@{##1}{##2}[##3]%
  \fi
}%
\AtEndDocument{\@gls@write@entrycounts}%
\renewcommand*{\@gls@entry@count}[2]{%
  \csgdef{glo@glsdetoklabel{##1}@prevcount}{##2}%
}%
\let\glsenableentrycount\relax
}
\@onlypreamble\glsenableentrycount
\newcommand*{\@gls@increment@currcount}[1]{%
  \csxdef{glo@glsdetoklabel{##1}@currcount}{%
    \number\numexpr\glsentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@local@increment@currcount}[1]{%
  \csedef{glo@glsdetoklabel{##1}@currcount}{%
    \number\numexpr\glsentrycurrcount{##1}+1}%
}
}

```

```

\newcommand*{\@gls@write@entrycounts}{%
  \immediate\write\@auxout
  {\string\providecommand*\string\@gls@entry@count}[2]{}}%
\forallglsentries{\@glsentry}{%
  \ifglsused{\@glsentry}%
  {\immediate\write\@auxout
   {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}}%
  }%
}%
}
\newcommand*{\@gls@entry@count}[2]{%
\newrobustcmd*{\cglS}{\@gls@hyp@opt\@cglS}
\newcommand*{\@cglS}[2][{}]{%
  \new@ifnextchar[{\@cglS@{#1}{#2}}{\@cglS@{#1}{#2}[]]}%
}
\def\@cglS@#1#2[#3]{%
  \GlossariesWarning{\string\cglS\space is defaulting to
    \string\gls\space since you haven't enabled entry counting}%
  \@cglS@{#1}{#2}[#3]}%
}
\newcommand*{\cglSformat}[2]{%
  \ifglshaslong{#1}{\glsentrylong{#1}}{\glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cGlS}{\@gls@hyp@opt\@cGlS}
\newcommand*{\@cGlS}[2][{}]{%
  \new@ifnextchar[{\@cGlS@{#1}{#2}}{\@cGlS@{#1}{#2}[]]}%
}
\def\@cGlS@#1#2[#3]{%
  \GlossariesWarning{\string\cGlS\space is defaulting to
    \string\Gls\space since you haven't enabled entry counting}%
  \@cGlS@{#1}{#2}[#3]}%
}
\newcommand*{\cGlSformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cglSpl}{\@gls@hyp@opt\@cglSpl}
\newcommand*{\@cglSpl}[2][{}]{%
  \new@ifnextchar[{\@cglSpl@{#1}{#2}}{\@cglSpl@{#1}{#2}[]]}%
}
\def\@cglSpl@#1#2[#3]{%
  \GlossariesWarning{\string\cglSpl\space is defaulting to
    \string\glspl\space since you haven't enabled entry counting}%
  \@cglSpl@{#1}{#2}[#3]}%
}
\newcommand*{\cglSplformat}[2]{%
  \ifglshaslong{#1}{\glsentrylongpl{#1}}{\glsentryfirstplural{#1}}#2%
}
\newrobustcmd*{\cGlSpl}{\@gls@hyp@opt\@cGlSpl}
\newcommand*{\@cGlSpl}[2][{}]{%
  \new@ifnextchar[{\@cGlSpl@{#1}{#2}}{\@cGlSpl@{#1}{#2}[]]}%
}

```

```

}
\def\cGlspl@#1#2[#3]{%
  \GlossariesWarning{\string\cGlspl\space is defaulting to
    \string\Glspl\space since you haven't enabled entry counting}%
  \@Glspl@{#1}{#2}[#3]%
}
\newcommand*\cGlsplformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
}
\newcommand*\loadglsentries}[2][\@gls@default]{%
  \let\@gls@default\glsdefaulttype
  \def\glsdefaulttype{#1}\input{#2}%
  \let\glsdefaulttype\@gls@default
}
\@onlypreamble{\loadglsentries}
\newcommand*\glsformat}[1]{#1}
\newcommand*\glsentryfmt{%
  \@@gls@default@entryfmt\glsdisplayfirst\glsdisplay
}
\newcommand*\@@gls@default@entryfmt}[2]{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          #2{\glsentryplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
          }%
        }%
        #1{\glsentryfirstplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
        }%
      }%
    }%
    \ifglsused\glslabel
    {%
      \ifbool{glscompatible-3.07}%
      {%
        \protected@edef\@glo@etext{%
          #2{\glsentryplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
          \xmakefirstuc\@glo@etext
        }%
      }%
    }%
  }%
}

```

```

#2{\Glsentryplural{\glslabel}}%
{\glsentrydescplural{\glslabel}}%
{\glsentrysymbolplural{\glslabel}}{\glsinsert}%
}%
}%
{%
\ifbool{glscompatible-3.07}%
{%
\protected@edef\@glo@etext{%
#1{\Glsentryfirstplural{\glslabel}}%
{\glsentrydescplural{\glslabel}}%
{\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
\xmakefirstuc\@glo@etext
}%
{%
#1{\Glsentryfirstplural{\glslabel}}%
{\glsentrydescplural{\glslabel}}%
{\glsentrysymbolplural{\glslabel}}{\glsinsert}%
}%
}%
}%
}%
\ifglsused\glslabel
{%
\mfirstucMakeUppercase{#2{\glsentryplural{\glslabel}}%
{\glsentrydescplural{\glslabel}}%
{\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
}%
{%
\mfirstucMakeUppercase{#1{\glsentryfirstplural{\glslabel}}%
{\glsentrydescplural{\glslabel}}%
{\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
}%
}%
}%
\glscapscase
{%
\ifglsused\glslabel
{%
#2{\glsentrytext{\glslabel}}%
{\glsentrydesc{\glslabel}}%
{\glsentrysymbol{\glslabel}}{\glsinsert}%
}%
{%
#1{\glsentryfirst{\glslabel}}%
{\glsentrydesc{\glslabel}}%
{\glsentrysymbol{\glslabel}}{\glsinsert}%
}%
}%

```

```

{%
  \ifglsused\glslabel
  {%
    \ifbool{glscompatible-3.07}%
    {%
      \protected@edef\@glo@etext{%
        #2{\glsentrytext{\glslabel}}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{\glsinsert}}%
      \xmakefirstuc\@glo@etext
    }%
    {%
      #2{\Glsentrytext{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
  }%
  {%
    \ifbool{glscompatible-3.07}%
    {%
      \protected@edef\@glo@etext{%
        #1{\glsentryfirst{\glslabel}}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{\glsinsert}}%
      \xmakefirstuc\@glo@etext
    }%
    {%
      #1{\Glsentryfirst{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
  }%
}
}
}
{%
  \ifglsused\glslabel
  {%
    \mfirstucMakeUppercase{#2{\glsentrytext{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}}%
  }%
  {%
    \mfirstucMakeUppercase{#1{\glsentryfirst{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}}%
  }%
}
}
}
{%
  \ifglsused{\glslabel}%

```



```

    {%
      #2{\glscustomtext}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{}}%
    }%
    {%
      #1{\glscustomtext}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{}}%
    }%
  }%
}
\newcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          \glsentryplural{\glslabel}\glsinsert
        }%
        {%
          \glsentryfirstplural{\glslabel}\glsinsert
        }%
      }%
    }%
    {%
      \ifglsused\glslabel
      {%
        \Glsentryplural{\glslabel}\glsinsert
      }%
      {%
        \Glsentryfirstplural{\glslabel}\glsinsert
      }%
    }%
    {%
      \ifglsused\glslabel
      {%
        \mfirstucMakeUppercase
        {\glsentryplural{\glslabel}\glsinsert}%
      }%
      {%
        \mfirstucMakeUppercase
        {\glsentryfirstplural{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glscapscase
  }%
}

```

```

    {%
      \ifglused\glslabel
      {%
        \glsentrytext{\glslabel}\glsinsert
      }%
      {%
        \glsentryfirst{\glslabel}\glsinsert
      }%
    }%
    {%
      \ifglused\glslabel
      {%
        \Glsentrytext{\glslabel}\glsinsert
      }%
      {%
        \Glsentryfirst{\glslabel}\glsinsert
      }%
    }%
    {%
      \ifglused\glslabel
      {%
        \mfirstucMakeUppercase{\glsentrytext{\glslabel}\glsinsert}%
      }%
      {%
        \mfirstucMakeUppercase{\glsentryfirst{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glscustomtext\glsinsert
  }%
}
\newcommand*{\glsnacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \acronymfont{\glsentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \mfirstucMakeUppercase
            {\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert}%
        }%
      }%
    }%
  }%
}

```



```

\newcommand*\genacrfullformat}[2]{%
  \glsentrylong{#1}#2\space
  (\protect\firstacronymfont{\glsentryshort{#1}})%
}
\newcommand*\Genacrfullformat}[2]{%
  \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
  \xmakefirstuc\gls@text
}
\newcommand*\genplacrfullformat}[2]{%
  \glsentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glsentryshortpl{#1}})%
}
\newcommand*\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \xmakefirstuc\gls@text
}
\newcommand*\glsdisplayfirst}[4]{#1#4}
\newcommand*\glsdisplay}[4]{#1#4}
\newcommand*\defglsdisplay}[2][\glsdefaulttype]{%
  \GlossariesWarning{\string\defglsdisplay\space is now obsolete.^^J
  Use \string\defglsentryfmt\space instead}%
  \expandafter\def\csname gls@#1@display\endcsname##1##2##3##4{#2}%
  \edef\@gls@doentrydef{%
    \noexpand\defglsentryfmt[#1]{%
      \noexpand\ifcsdef{gls@#1@displayfirst}%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\glsdisplayfirst}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
    }%
  }%
  \@gls@doentrydef
}
\newcommand*\defglsdisplayfirst}[2][\glsdefaulttype]{%
  \GlossariesWarning{\string\defglsdisplayfirst\space is now obsolete.^^J
  Use \string\defglsentryfmt\space instead}%
  \expandafter\def\csname gls@#1@displayfirst\endcsname##1##2##3##4{#2}%
  \edef\@gls@doentrydef{%
    \noexpand\defglsentryfmt[#1]{%
      \noexpand\ifcsdef{gls@#1@display}%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
    }%
  }%
  \@gls@doentrydef
}

```

```

    }%
    {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\glsdisplay}%
    }%
    }%
    }%
    \@gls@doentrydef
}
\define@key{glslink}{counter}{%
    \ifcsundef{c@#1}%
    {%
        \PackageError{glossaries}%
        {There is no counter called ‘#1’}%
        {%
            The counter key should have the name of a valid counter
            as its value%
        }%
    }%
    }%
    {%
        \def\@gls@counter{#1}%
    }%
}
\define@key{glslink}{format}{%
    \def\@glsnumberformat{#1}}
\define@boolkey{glslink}{hyper}[true]{}
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\define@boolkey{glslink}{local}[true]{}
\newcommand*{\glslinkvar}[3]{#1}
\newcommand*{\glsifhyper}[2]{%
    \glslinkvar{#1}{#2}{#1}%
    \GlossariesWarning{\string\glsifhyper\space is deprecated. Did
        you mean \string\glsifhyperon\space or \string\glslinkvar?}%
}
\newcommand*{\@gls@hyp@opt}[1]{%
    \let\glslinkvar\@firstofthree
    \let\@gls@hyp@opt@cs#1\relax
    \@ifstar{\s@gls@hyp@opt}%
    {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*{\s@gls@hyp@opt}[1][]{%
    \let\glslinkvar\@secondofthree
    \@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1][]{%
    \let\glslinkvar\@thirdofthree
    \@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
    \@gls@hyp@opt\@gls@@link
}

```

```

\newcommand*{\@gls@link}[3][\%
  \glsdoifexistsordo{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \@gls@link[#1]{#2}{#3}%
  }{%
    \glstextformat{#3}%
  }%
  \glspostlinkhook
}
\newcommand*{\glspostlinkhook}{}
\newcommand*{\@gls@link@checkfirsthyper}{%
  \ifglsused{\glslabel}%
  {%
  }%
  {%
    \gls@checkisacronymlist\glstype
    \ifglshyperfirst
      \ifglsisacronymlist
        \ifglsacrfootnote
          \KV@glslink@hyperfalse
        \fi
      \fi
    \else
      \KV@glslink@hyperfalse
    \fi
  }%
  \glslinkcheckfirsthyperhook
}
\newcommand*{\glslinkcheckfirsthyperhook}{}
\newcommand*{\glslinkpostsetkeys}{}
\newcommand{\@glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}
\newcommand*{\do@glsdisablehyperinlist}{%
  \expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
  {\KV@glslink@hyperfalse}{}%
}
\newcommand*{\@gls@setdefault@glslink@opts}{}
\def\@gls@link[#1]#2#3{%
  \leavevmode
  \edef\glslabel{\glsdetoklabel{#2}}%
  \def\@gls@link@opts{#1}%
  \let\@gls@link@label\glslabel
  \def\@glsnumberformat{glsnumberformat}%
  \edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  \edef\glstype{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
  \@gls@setdefault@glslink@opts
  \do@glsdisablehyperinlist
  \do@gls@link@checkfirsthyper
  \setkeys{glslink}{#1}%
}

```

```

\glslinkpostsetkeys
\@gls@saveentrycounter
\@gls@setsort{\glslabel}%
\@do@wrglossary{#2}%
\ifKV@glslink@hyper
  \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
\else
  \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
\fi
\let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
}
\newcommand*{\glolinkprefix}{glo:}
\def\glsentrycounter{\glscounter}%
\newcommand*{\@gls@saveentrycounter}{%
  \def\@gls@Hcounter{}%
  \ifthenelse{\equal{\@gls@counter}{equation}}%
  {
    \ifcsundef{xatlevel@}%
    {%
      \edef\theglsentrycounter{\expandafter\noexpand
        \csname the\@gls@counter\endcsname}%
    }%
    {%
      \ifx\xatlevel@\@empty
        \edef\theglsentrycounter{\expandafter\noexpand
          \csname the\@gls@counter\endcsname}%
      \else
        \savecounters@
        \advance\c@equation by 1\relax
        \edef\theglsentrycounter{\csname the\@gls@counter\endcsname}%
        \ifcsundef{theH\@gls@counter}%
        {%
          \def\@gls@Hcounter{\theglsentrycounter}%
        }%
        {%
          \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
        }%
        \protected@edef\theHglentrycounter{\@gls@Hcounter}%
        \restorecounters@
      \fi
    }%
  }%
  {%
    \edef\theglsentrycounter{\expandafter\noexpand
      \csname the\@gls@counter\endcsname}%
  }%
  \ifx\@gls@Hcounter\@empty
    \ifcsundef{theH\@gls@counter}%
    {%
      \def\theHglentrycounter{\theglsentrycounter}%
    }%
  }%

```

```

    }%
    {%
        \protected@edef\theHglentrycounter{\expandafter\noexpand
            \csname theH\@gls@counter\endcsname}%
    }%
\fi
}
\def\@set@glo@numformat#1#2#3#4{%
    \expandafter\@glo@check@mkidxrangear#3\@nil
    \protected@edef#1{%
        \glo@prefix setentrycounter[#4]{#2}%
        \expandafter\string\csname\@glo@suffix\endcsname
    }%
    \@gls@checkmkidxchars#1%
}
\def\@glo@check@mkidxrangear#1#2\@nil{%
\if#1\relax
    \def\@glo@prefix{()}%
    \if\relax#2\relax
        \def\@glo@suffix{glsnumberformat}%
    \else
        \def\@glo@suffix{#2}%
    \fi
\else
    \if#1\relax
        \def\@glo@prefix{}%
        \if\relax#2\relax
            \def\@glo@suffix{glsnumberformat}%
        \else
            \def\@glo@suffix{#2}%
        \fi
    \else
        \def\@glo@prefix{}\def\@glo@suffix{#1#2}%
    \fi
\fi}
\newcommand*{\@gls@escbsdq}[1]{%
    \def\@gls@checkedmkidx{}%
    \let\gls@xdystring=#1\relax
    \@onelevel@sanitize\gls@xdystring
    \edef\do@gls@xdycheckbackslash{%
        \noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
        \@backslashchar\@backslashchar\noexpand\null}%
    \do@gls@xdycheckbackslash
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
    \@for\@gls@tmp:=\gls@protected@pagefmts\do
    {%
        \edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\ \expandonce\@gls@tmp}%
    }
}

```



```

    \@onelevel@sanitize\@gls@sanitized@tmp
    \edef\gls@dostsubst{%
      \noexpand\DTLsubstituteall\noexpand\gls@xdystring
      {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
    }%
    \gls@dostsubst
  }%
  \let#1=\gls@xdystring
}
\newcommand{\@gls@checkmkidxchars}[1]{%
  \ifglsxindy
    \@gls@escbsdq{#1}%
  \else
    \def\@gls@checkedmkidx{%
      \expandafter\@gls@checkquote#1\@nil""\null
      \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
      \def\@gls@checkedmkidx{%
        \expandafter\@gls@checkescquote#1\@nil\\"\null
        \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
        \def\@gls@checkedmkidx{%
          \expandafter\@gls@checkescactual#1\@nil\?\?\null
          \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
          \def\@gls@checkedmkidx{%
            \expandafter\@gls@checkactual#1\@nil??\null
            \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
            \def\@gls@checkedmkidx{%
              \expandafter\@gls@checkbar#1\@nil||\null
              \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
              \def\@gls@checkedmkidx{%
                \expandafter\@gls@checkescbar#1\@nil|||\null
                \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                \def\@gls@checkedmkidx{%
                  \expandafter\@gls@checklevel#1\@nil!!\null
                  \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                }%
              }%
            }%
          }%
        }%
      }%
    }%
  \fi
}
\def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
\newtoks\@gls@tmpb
\def\@gls@checkquote#1"#2"#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\@nil#2\null
    \ifx\@nil#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkquote{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
      \def\@gls@checkquote{\@gls@checkquote#3\null}%
    \fi
  \fi
}

```

```

\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@quotechar}%
\ifx\@null#3\@null
\def\@@gls@checkquote{\@gls@checkquote#2""\@null}%
\else
\def\@@gls@checkquote{\@gls@checkquote#2"#3\@null}%
\fi
\fi
\fi
\@@gls@checkquote
}
\def\@gls@checkescquote#1"#2"#3\@null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\@null#2\@null
\ifx\@null#3\@null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkescquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar
\@gls@quotechar\string"\@gls@quotechar}%
\def\@@gls@checkescquote{\@gls@checkescquote#3\@null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar}%
\ifx\@null#3\@null
\def\@@gls@checkescquote{\@gls@checkescquote#2""\@null}%
\else
\def\@@gls@checkescquote{\@gls@checkescquote#2"#3\@null}%
\fi
\fi
\fi
\@@gls@checkescquote
}
\def\@gls@checkescactual#1\?#2\?#3\@null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\@null#2\@null
\ifx\@null#3\@null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkescactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar
\@gls@quotechar\string"\@gls@actualchar}%
\def\@@gls@checkescactual{\@gls@checkescactual#3\@null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@

```

```

\@gls@quotechar\string\ "@gls@actualchar}%
\ifx\null#3\null
\def\@gls@checkescactual{\@gls@checkescactual#2\?\?\null}%
\else
\def\@gls@checkescactual{\@gls@checkescactual#2\?#3\null}%
\fi
\fi
\@gls@checkescactual
}
\def\@gls@checkescbar#1\|#2\|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\ "@gls@encapchar
\@gls@quotechar\string\ "@gls@encapchar}%
\def\@gls@checkescbar{\@gls@checkescbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\ "@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkescbar{\@gls@checkescbar#2\|\|\null}%
\else
\def\@gls@checkescbar{\@gls@checkescbar#2\|#3\null}%
\fi
\fi
\@gls@checkescbar
}
\def\@gls@checkesclevel#1\!#2\!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkesclevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\ "@gls@levelchar
\@gls@quotechar\string\ "@gls@levelchar}%
\def\@gls@checkesclevel{\@gls@checkesclevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\ "@gls@levelchar}%
\ifx\null#3\null

```

```

\def\@gls@checkesclevel{\@gls@checkesclevel#2\!\!\null}%
\else
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!#3\null}%
\fi
\fi
\@gls@checkesclevel
}
\def\@gls@checkbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@gls@checkbar{\@gls@checkbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkbar{\@gls@checkbar#2||\null}%
\else
\def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}
\def\@gls@checklevel#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checklevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
\def\@gls@checklevel{\@gls@checklevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checklevel{\@gls@checklevel#2!!\null}%
\else
\def\@gls@checklevel{\@gls@checklevel#2|#3\null}%
\fi

```

```

\fi
\@gls@checklevel
}
\def\@gls@checkactual#1?#2?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
\def\@gls@checkactual{\@gls@checkactual#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar}%
\ifx\null#3\null
\def\@gls@checkactual{\@gls@checkactual#2??\null}%
\else
\def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
\fi
\fi
\@gls@checkactual
}
\def\@gls@xdycheckquote#1"#2"#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@xdycheckquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\string"\string"}%
\def\@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\string"}%
\ifx\null#3\null
\def\@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
\else
\def\@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
\fi
\fi
\@gls@xdycheckquote
}
\edef\def@gls@xdycheckbackslash{%

```

```

\newcommand*{\gls@xdycheckbackslash##1\@backslashchar
##2\@backslashchar##3\@backslashchar}{%
\newcommand*{\gls@tmpb}=\noexpand\expandafter
{\noexpand\@gls@checkedmkidx}%
\newcommand*{\toks@}={##1}%
\newcommand*{\ifx\@backslashchar##2\@backslashchar}
\newcommand*{\ifx\@backslashchar##3\@backslashchar}
\newcommand*{\edef\@gls@checkedmkidx{%
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}}%
\newcommand*{\the\@gls@tmpb}=\noexpand\@gls@checkedmkidx{%
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}}%
\newcommand*{\relax}{}%
\newcommand*{\else}{}%
\newcommand*{\edef\@gls@checkedmkidx{%
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}}%
\newcommand*{\def\@gls@xdycheckbackslash{%
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}}%
\newcommand*{\fi}{}%
\newcommand*{\else}{}%
\newcommand*{\edef\@gls@checkedmkidx{%
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}}%
\newcommand*{\ifx\@backslashchar##3\@backslashchar}
\newcommand*{\def\@gls@xdycheckbackslash{%
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}}%
\newcommand*{\else}{}%
\newcommand*{\def\@gls@xdycheckbackslash{%
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}}%
\newcommand*{\fi}{}%
\newcommand*{\fi}{}%
\newcommand*{\@gls@xdycheckbackslash}{}%
}%
}
\def\gls@xdycheckbackslash

\newlength\gls@tmplen
\newcommand*{\glsdohypertarget}[2]{%
\@gls@showtarget{##1}%
\settoheight\gls@tmplen{##2}%
\raisebox{\gls@tmplen}{\hypertarget{##1}{}}{##2}%
}
\newcommand*{\glsdohyperlink}[2]{%
\@gls@showtarget{##1}%
\hyperlink{##1}{##2}%
}
\newcommand*{\glsdonohyperlink}[2]{##2}
\ifcsundef{hyperlink}%
{%
\let\glslink\glsdonohyperlink

```

```

}%
{%
  \let\@glslink\glsdohyperlink
}
\ifcsundef{hypertarget}%
{%
  \let\@glstarget\@secondoftwo
}%
{%
  \let\@glstarget\glsdohypertarget
}
\newcommand{\glsdisablehyper}{%
  \KV@glslink@hyperfalse
  \let\@glslink\glsdonohyperlink
  \let\@glstarget\@secondoftwo
}
\newcommand{\glsenablehyper}{%
  \KV@glslink@hypertrue
  \let\@glslink\glsdohyperlink
  \let\@glstarget\glsdohypertarget
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}
\newcommand*{\@gls}[2] [] {%
  \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2} []}]%
}
\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glsescapscase\@firstofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
\newcommand*{\@Gls}[2] [] {%
  \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2} []}]%
}
\def\@Gls@#1#2[#3]{%

```

```

\glsdoifexists{#2}%
{%
  \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
  \let\glsifplural\@secondoftwo
  \let\glscapscase\@secondofthree
  \let\glscustomtext\@empty
  \def\glsinsert{#3}%
  \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
  \@gl@link[#1]{#2}{\@glo@text}%
  \ifKV@gl@link@local
    \glslocalunset{#2}%
  \else
    \glsunset{#2}%
  \fi
}%
\glspostlinkhook
}
\newrobustcmd*{\GLS}{\@gl@hyp@opt\@GLS}
\newcommand*{\@GLS}[2] []{%
  \new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2} []}%
}
\def\@GLS@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \ifKV@gl@link@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\glspl}{\@gl@hyp@opt\@glspl}
\newcommand*{\@glspl}[2] []{%
  \new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2} []}%
}
\def\@glspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glscustomtext\@empty

```



```

\def\glsinsert{#3}%
\def\glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\ifKV@glslink@local
  \glslocalunset{#2}%
\else
  \glsunset{#2}%
\fi
}%
\glspostlinkhook
}
\newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
\newcommand*{\@Glspl}[2][{}]{%
  \new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2}[]}%
}
\def\@Glspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\gls@scaps@case\@secondofthree
    \let\gls@custom@text\@empty
    \def\glsinsert{#3}%
    \def\glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLSp1}{\@gls@hyp@opt\@GLSp1}
\newcommand*{\@GLSp1}[2][{}]{%
  \new@ifnextchar[{\@GLSp1@{#1}{#2}}{\@GLSp1@{#1}{#2}[]}%
}
\def\@GLSp1@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\gls@scaps@case\@thirdofthree
    \let\gls@custom@text\@empty
    \def\glsinsert{#3}%
    \def\glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else

```

```

        \glsunset{#2}%
    \fi
}%
\glspostlinkhook
}
\newrobustcmd*{\glsdisp}{\@gls@hyp@opt\@glsdisp}
\newcommand*{\@glsdisp}[3] []{%
    \glsdoifexists{#2}{%
        \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
        \let\glsifplural\@secondoftwo
        \let\glsescapscase\@firstofthree
        \def\glscustomtext{#3}%
        \def\glsinsert{}%
        \def\glo@text{\csname gls@glstype @entryfmt\endcsname}%
        \@gls@link{#1}{#2}{\@glo@text}%
        \ifKV@glslink@local
            \glslocalunset{#2}%
        \else
            \glsunset{#2}%
        \fi
    }%
    \glspostlinkhook
}
\newcommand*{\@gls@link@nocheckfirsthyper}{}
\newcommand{\@gls@field@link}[3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \@gls@link{#1}{#2}{#3}%
    }%
    \glspostlinkhook
}
\newrobustcmd*{\glsstext}{\@gls@hyp@opt\@glsstext}
\newcommand*{\@glsstext}[2] []{%
    \new@ifnextchar[{\@glsstext@{#1}{#2}}{\@glsstext@{#1}{#2} []}]
\def\@glsstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentrytext{#2}#3}%
}
\newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
\newcommand*{\@GLStext}[2] []{%
    \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2} []}]
\def\@GLStext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrytext{#2}#3}}%
}
\newrobustcmd*{\Glstext}{\@gls@hyp@opt\@Glstext}
\newcommand*{\@Glstext}[2] []{%
    \new@ifnextchar[{\@Glstext@{#1}{#2}}{\@Glstext@{#1}{#2} []}]
\def\@Glstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentrytext{#2}#3}%
}
}

```

```

\newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
\newcommand*{\@glsfirst}[2] [] {%
  \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2} []}]
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
}
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
\newcommand*{\@Glsfirst}[2] [] {%
  \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2} []}]
\def\@Glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
\newcommand*{\@GLSfirst}[2] [] {%
  \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2} []}]
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirst{#2}#3}}%
}
\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
\newcommand*{\@glsplural}[2] [] {%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} []}]
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\newcommand*{\@Glsplural}[2] [] {%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} []}]
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
\newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
\newcommand*{\@GLSplural}[2] [] {%
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} []}]
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryplural{#2}#3}}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{\@glsfirstplural}[2] [] {%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} []}]
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
\newcommand*{\@Glsfirstplural}[2] [] {%
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2} []}]
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
\newcommand*{\@GLSfirstplural}[2] [] {%

```

```

\new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} []}]
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirstplural{#2}#3}}%
}
\newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
\newcommand*{\@glsname}[2] []{%
  \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2} []}]
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryname{#2}#3}%
}
\newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}
\newcommand*{\@Glsname}[2] []{%
  \new@ifnextchar[{\@Glsname@{#1}{#2}}{\@Glsname@{#1}{#2} []}]
\def\@Glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryname{#2}#3}%
}
\newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}
\newcommand*{\@GLSname}[2] []{%
  \new@ifnextchar[{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}]
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryname{#2}#3}}%
}
\newrobustcmd*{\glsdesc}{\@gls@hyp@opt\@glsdesc}
\newcommand*{\@glsdesc}[2] []{%
  \new@ifnextchar[{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2} []}]
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydesc{#2}#3}%
}
\newrobustcmd*{\Glsdesc}{\@gls@hyp@opt\@Glsdesc}
\newcommand*{\@Glsdesc}[2] []{%
  \new@ifnextchar[{\@Glsdesc@{#1}{#2}}{\@Glsdesc@{#1}{#2} []}]
\def\@Glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%
}
\newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\newcommand*{\@GLSdesc}[2] []{%
  \new@ifnextchar[{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}]
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydesc{#2}#3}}%
}
\newrobustcmd*{\glsdescplural}{\@gls@hyp@opt\@glsdescplural}
\newcommand*{\@glsdescplural}[2] []{%
  \new@ifnextchar[{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}]
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\Glsdescplural}{\@gls@hyp@opt\@Glsdescplural}
\newcommand*{\@Glsdescplural}[2] []{%
  \new@ifnextchar[{\@Glsdescplural@{#1}{#2}}{\@Glsdescplural@{#1}{#2} []}]
\def\@Glsdescplural@#1#2[#3]{%

```

```

    \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\newcommand*{\@GLSdescplural}[2] [] {%
    \new@ifnextchar [{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}]
\def \@GLSdescplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydescplural{#2}#3}}%
}
\newrobustcmd*{\glssymbol}{\@gls@hyp@opt\@glssymbol}
\newcommand*{\@glssymbol}[2] [] {%
    \new@ifnextchar [{\@glssymbol@{#1}{#2}}{\@glssymbol@{#1}{#2} []}]
\def \@glssymbol@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\Glsymbol}{\@gls@hyp@opt\@Glsymbol}
\newcommand*{\@Glsymbol}[2] [] {%
    \new@ifnextchar [{\@Glsymbol@{#1}{#2}}{\@Glsymbol@{#1}{#2} []}]
\def \@Glsymbol@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\GLSsymbol}{\@gls@hyp@opt\@GLSsymbol}
\newcommand*{\@GLSsymbol}[2] [] {%
    \new@ifnextchar [{\@GLSsymbol@{#1}{#2}}{\@GLSsymbol@{#1}{#2} []}]
\def \@GLSsymbol@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrysymbol{#2}#3}}%
}
\newrobustcmd*{\glssymbolplural}{\@gls@hyp@opt\@glssymbolplural}
\newcommand*{\@glssymbolplural}[2] [] {%
    \new@ifnextchar [{\@glssymbolplural@{#1}{#2}}{\@glssymbolplural@{#1}{#2} []}]
\def \@glssymbolplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\Glsymbolplural}{\@gls@hyp@opt\@Glsymbolplural}
\newcommand*{\@Glsymbolplural}[2] [] {%
    \new@ifnextchar [{\@Glsymbolplural@{#1}{#2}}{\@Glsymbolplural@{#1}{#2} []}]
\def \@Glsymbolplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrysymbolplural{#2}#3}}%
}
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
\newcommand*{\@glsuseri}[2] [] {%
    \new@ifnextchar [{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2} []}]
\def \@glsuseri@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsentryuseri{#2}#3}%
}
}

```

```

\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
\newcommand*{\@Glsuseri}[2] [] {%
  \new@ifnextchar[{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2} []}]
\def\@Glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\newcommand*{\@GLSuseri}[2] [] {%
  \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2} []}]
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseri{#2}#3}}%
}
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
\newcommand*{\@glsuserii}[2] [] {%
  \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2} []}]
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
\newcommand*{\@Glsuserii}[2] [] {%
  \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2} []}]
\def\@Glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\GLSuserii}{\@gls@hyp@opt\@GLSuserii}
\newcommand*{\@GLSuserii}[2] [] {%
  \new@ifnextchar[{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2} []}]
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuserii{#2}#3}}%
}
\newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}
\newcommand*{\@glsuseriii}[2] [] {%
  \new@ifnextchar[{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2} []}]
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
\newcommand*{\@Glsuseriii}[2] [] {%
  \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2} []}]
\def\@Glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseriii{#2}#3}}%
}
\newrobustcmd*{\glsuseriv}{\@gls@hyp@opt\@glsuseriv}
\newcommand*{\@glsuseriv}[2] [] {%

```

```

    \new@ifnextchar [ {\@glsuseriv@{#1}{#2}} {\@glsuseriv@{#1}{#2} [] }
\def \@glsuseriv@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\glsentryuseriv{#2}#3}%
}
\newrobustcmd* {\Glsuseriv} {\@gls@hyp@opt \@Glsuseriv}
\newcommand* {\@Glsuseriv}[2] [] {%
    \new@ifnextchar [ {\@Glsuseriv@{#1}{#2}} {\@Glsuseriv@{#1}{#2} [] }
\def \@Glsuseriv@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\Glsentryuseriv{#2}#3}%
}
\newrobustcmd* {\GLSuseriv} {\@gls@hyp@opt \@GLSuseriv}
\newcommand* {\@GLSuseriv}[2] [] {%
    \new@ifnextchar [ {\@GLSuseriv@{#1}{#2}} {\@GLSuseriv@{#1}{#2} [] }
\def \@GLSuseriv@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\mfirstucMakeUppercase{\glsentryuseriv{#2}#3}}%
}
\newrobustcmd* {\glsuserv} {\@gls@hyp@opt \@glsuserv}
\newcommand* {\@glsuserv}[2] [] {%
    \new@ifnextchar [ {\@glsuserv@{#1}{#2}} {\@glsuserv@{#1}{#2} [] }
\def \@glsuserv@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\glsentryuserv{#2}#3}%
}
\newrobustcmd* {\Glsuserv} {\@gls@hyp@opt \@Glsuserv}
\newcommand* {\@Glsuserv}[2] [] {%
    \new@ifnextchar [ {\@Glsuserv@{#1}{#2}} {\@Glsuserv@{#1}{#2} [] }
\def \@Glsuserv@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\mfirstucMakeUppercase{\glsentryuserv{#2}#3}}%
}
\newrobustcmd* {\glsuservi} {\@gls@hyp@opt \@glsuservi}
\newcommand* {\@glsuservi}[2] [] {%
    \new@ifnextchar [ {\@glsuservi@{#1}{#2}} {\@glsuservi@{#1}{#2} [] }
\def \@glsuservi@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\glsentryuservi{#2}#3}%
}
\newrobustcmd* {\GLSuservi} {\@gls@hyp@opt \@GLSuservi}
\newcommand* {\@GLSuservi}[2] [] {%
    \new@ifnextchar [ {\@GLSuservi@{#1}{#2}} {\@GLSuservi@{#1}{#2} [] }
\def \@GLSuservi@#1#2[#3] {%

```

```

\@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuservi{#2}#3}}%
}
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\ns@acrshort}
\newcommand*{\ns@acrshort}[2] [] {%
  \new@ifnextchar[{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2} []}]%
}
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrshort}{\@gls@hyp@opt\ns@Acrshort}
\newcommand*{\ns@Acrshort}[2] [] {%
  \new@ifnextchar[{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2} []}]%
}
\def\@Acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshort}{\@gls@hyp@opt\ns@ACRshort}
\newcommand*{\ns@ACRshort}[2] [] {%
  \new@ifnextchar[{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}]%
}
\def\@ACRshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsapscase\@thirdofthree

```



```

\let\glsinsert\@empty
\def\glscustomtext{%
  \mfirstucMakeUppercase{\acronymfont{\glsentryshort{#2}}#3}%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\acrshortpl}{\@gls@hyp@opt\ns@acrshortpl}
\newcommand*{\ns@acrshortpl}[2] [] {%
  \new@ifnextchar[{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} []}%
}
\def\@acrshortpl#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\gls caps case\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\ns@Acrshortpl}
\newcommand*{\ns@Acrshortpl}[2] [] {%
  \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2} []}%
}
\def\@Acrshortpl#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\gls caps case\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\ns@ACRshortpl}
\newcommand*{\ns@ACRshortpl}[2] [] {%
  \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2} []}%
}

```

```

\def\@ACRshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{#2}}#3}%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrlong}{\@gl@hyp@opt\ns@acrlong}
\newcommand*{\ns@acrlong}[2][{}]{%
  \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2}[]}%
}
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsentrylong{#2}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrlong}{\@gl@hyp@opt\ns@Acrlong}
\newcommand*{\ns@Acrlong}[2][{}]{%
  \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2}[]}%
}
\def\@Acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylong{#2}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%

```

```

}%
\glspostlinkhook
}
\newrobustcmd*{\ACRlong}{\@gls@hyp@opt\ns@ACRlong}
\newcommand*{\ns@ACRlong}[2] [] {%
  \new@ifnextchar [{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2} []}%
}
\def\@ACRlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\glsentrylong{#2}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\ns@acrlongpl}
\newcommand*{\ns@acrlongpl}[2] [] {%
  \new@ifnextchar [{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2} []}%
}
\def\@acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\ns@Acrlongpl}
\newcommand*{\ns@Acrlongpl}[2] [] {%
  \new@ifnextchar [{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2} []}%
}
\def\@Acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%

```

```

\let\glsifplural\@firstoftwo
\let\glscapscase\@secondofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \Glsentrylongpl{#2}#3%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\@ns@ACRlongpl}
\newcommand*{\ns@ACRlongpl}[2][{}]{%
  \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2}[]}%
}
\def\@ACRlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glsapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\glsentrylongpl{#2}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newcommand*{\@gls@entry@field}[2]{%
  \csname glo@\glsdetoklabel{#1}@\#2\endcsname
}
\newcommand*{\glsletentryfield}[3]{%
  \letcs{#1}{glo@\glsdetoklabel{#2}@\#3}%
}
\newcommand*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@\#2}%
    \ifdef\@glo@text
    {%
      \xmakefirstuc{\@glo@text}%
    }%
  }%
  ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
  entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
  label and the field name}%
}%
}%

```

```

    ???%
  }%
}
\newcommand*\glsentryname}[1]{\@gls@entry@field{#1}{name}}
\newrobustcmd*\Glsentryname}[1]{%
  \@Gls@entryname{#1}%
}
\newcommand*\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
\newcommand*\@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo\@glsdetoklabel{#1}@name}%
    \expandafter\@gls@getbody\@glo@text{}\@nil
    \expandafter\ifx\@gls@body\glsentrylong\relax
    \expandafter\Glsentrylong\@gls@rest
  \else
    \expandafter\ifx\@gls@body\glsentryshort\relax
    \expandafter\Glsentryshort\@gls@rest
  \else
    \expandafter\ifx\@gls@body\acronymfont\relax
    {%
      \let\glsentryshort\Glsentryshort
      \@glo@text
    }%
  \else
    \expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
    {%
      \let\glsentryshort\Glsentryshort
      \@glo@text
    }%
  \else
    \xmakefirstuc{\@glo@text}%
  \fi
  \fi
  \fi
  }%
  {%
    \@Gls@entry@field{#1}{name}%
  }%
}
\newcommand*\glsentrydesc}[1]{\@gls@entry@field{#1}{desc}}
\newrobustcmd*\Glsentrydesc}[1]{%
  \@Gls@entry@field{#1}{desc}%
}
\newcommand*\glsentrydescplural}[1]{%
  \@gls@entry@field{#1}{descplural}%
}
}

```

```

\newrobustcmd*{\Glsentrydescplural}[1]{%
  \@Gls@entry@field{#1}{descplural}%
}
\newcommand*{\glsentrytext}[1]{\@gls@entry@field{#1}{text}}
\newrobustcmd*{\Glsentrytext}[1]{%
  \@Gls@entry@field{#1}{text}%
}
\newcommand*{\glsentryplural}[1]{%
  \@gls@entry@field{#1}{plural}%
}
\newrobustcmd*{\Glsentryplural}[1]{%
  \@Gls@entry@field{#1}{plural}%
}
\newcommand*{\glsentrysymbol}[1]{%
  \@gls@entry@field{#1}{symbol}%
}
\newrobustcmd*{\Glsentrysymbol}[1]{%
  \@Gls@entry@field{#1}{symbol}%
}
\newcommand*{\glsentrysymbolplural}[1]{%
  \@gls@entry@field{#1}{symbolplural}%
}
\newrobustcmd*{\Glsentrysymbolplural}[1]{%
  \@Gls@entry@field{#1}{symbolplural}%
}
\newcommand*{\glsentryfirst}[1]{%
  \@gls@entry@field{#1}{first}%
}
\newrobustcmd*{\Glsentryfirst}[1]{%
  \@Gls@entry@field{#1}{first}%
}
\newcommand*{\glsentryfirstplural}[1]{%
  \@gls@entry@field{#1}{firstpl}%
}
\newrobustcmd*{\Glsentryfirstplural}[1]{%
  \@Gls@entry@field{#1}{firstpl}%
}
\newrobustcmd*{\@glsentrytitlecase}[2]{%
  \glsdoifexists{#1}%
  {%
    \glsfieldfetch{#1}{#2}{\@gls@value}%
    \xcapitalisewords{\@gls@value}%
  }%
}
\ifdef\texorpdfstring
{
  \newcommand*{\glsentrytitlecase}[2]{%
    \texorpdfstring
      {\@glsentrytitlecase{#1}{#2}}%
      {\@gls@entry@field{#1}{#2}}%
}

```

```

    }
  }
  {
    \newcommand*\glentrytitlecase}[2]{\@glentrytitlecase{#1}{#2}}
  }
  \newcommand*\glentrytype}[1]{\@gls@entry@field{#1}{type}}
  \newcommand*\glentrysort}[1]{%
    \@gls@entry@field{#1}{sort}%
  }
  \newcommand*\glentryparent}[1]{%
    \@gls@entry@field{#1}{parent}%
  }
  \newcommand*\glentryuseri}[1]{%
    \@gls@entry@field{#1}{useri}%
  }
  \newrobustcmd*\Glsentryuseri}[1]{%
    \@Gls@entry@field{#1}{useri}%
  }
  \newcommand*\glentryuserii}[1]{%
    \@gls@entry@field{#1}{userii}%
  }
  \newrobustcmd*\Glsentryuserii}[1]{%
    \@Gls@entry@field{#1}{userii}%
  }
  \newcommand*\glentryuseriii}[1]{%
    \@gls@entry@field{#1}{useriii}%
  }
  \newrobustcmd*\Glsentryuseriii}[1]{%
    \@Gls@entry@field{#1}{useriii}%
  }
  \newcommand*\glentryuseriv}[1]{%
    \@gls@entry@field{#1}{useriv}%
  }
  \newrobustcmd*\Glsentryuseriv}[1]{%
    \@Gls@entry@field{#1}{useriv}%
  }
  \newcommand*\glentryuserv}[1]{%
    \@gls@entry@field{#1}{userv}%
  }
  \newrobustcmd*\Glsentryuserv}[1]{%
    \@Gls@entry@field{#1}{userv}%
  }
  \newcommand*\glentryuservi}[1]{%
    \@gls@entry@field{#1}{uservi}%
  }
  \newrobustcmd*\Glsentryuservi}[1]{%
    \@Gls@entry@field{#1}{uservi}%
  }
  \newcommand*\glentryshort}[1]{\@gls@entry@field{#1}{short}}
  \newrobustcmd*\Glsentryshort}[1]{%

```

```

    \@Gls@entry@field{#1}{short}%
  }
  \newcommand*\glentryshortpl}[1]{\@Gls@entry@field{#1}{shortpl}}
  \newrobustcmd*\Glsentryshortpl}[1]{%
    \@Gls@entry@field{#1}{shortpl}%
  }
  \newcommand*\glentrylong}[1]{\@Gls@entry@field{#1}{long}}
  \newrobustcmd*\Glsentrylong}[1]{%
    \@Gls@entry@field{#1}{long}%
  }
  \newcommand*\glentrylongpl}[1]{\@Gls@entry@field{#1}{longpl}}
  \newrobustcmd*\Glsentrylongpl}[1]{%
    \@Gls@entry@field{#1}{longpl}%
  }
  \newcommand*\glentryfull}[1]{%
    \acrfullformat{\glentrylong{#1}}{\acronymfont{\glentryshort{#1}}}%
  }
  \newrobustcmd*\Glsentryfull}[1]{%
    \acrfullformat{\Glsentrylong{#1}}{\acronymfont{\glentryshort{#1}}}%
  }
  \newcommand*\glentryfullpl}[1]{%
    \acrfullformat{\glentrylongpl{#1}}{\acronymfont{\glentryshortpl{#1}}}%
  }
  \newrobustcmd*\Glsentryfullpl}[1]{%
    \acrfullformat{\Glsentrylongpl{#1}}{\acronymfont{\glentryshortpl{#1}}}%
  }
  \newcommand*\glentrynumberlist}[1]{%
    \glsdoifexists{#1}%
    {%
      \@Gls@entry@field{#1}{numberlist}%
    }%
  }
  \ifpackageloaded{hyperref} {%
    \newcommand*\glsdisplaynumberlist}[1]{%
      \GlossariesWarning
      {%
        \string\glsdisplaynumberlist\space
        doesn't work with hyperref.^^JUsing
        \string\glentrynumberlist\space instead%
      }%
      \glentrynumberlist{#1}%
    }%
  }%
  {%
    \newcommand*\glsdisplaynumberlist}[1]{%
      \glsdoifexists{#1}%
      {%
        \bgroup
          \edef\@glo@label{\glsdetoklabel{#1}}%
          \let\@org@glsnumberformat\glsnumberformat

```



```

\def\glsnumberformat##1{##1}%
\protected@edef\the@numberlist{%
  \csname glo@\@glo@label @numberlist\endcsname}%
\def\@gls@numlist@sep{}%
\def\@gls@numlist@nextsep{}%
\def\@gls@numlist@lastsep{}%
\def\@gls@thislist{}%
\def\@gls@donext@def{}%
\renewcommand\do[1]{%
  \protected@edef\@gls@thislist{%
    \@gls@thislist
    \noexpand\@gls@numlist@sep
    ##1%
  }%
  \let\@gls@numlist@sep\@gls@numlist@nextsep
  \def\@gls@numlist@nextsep{\glsnumlistsep}%
  \@gls@donext@def
  \def\@gls@donext@def{%
    \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
  }%
}%
\expandafter \glsnumlistparser \expandafter{\the@numberlist}%
\let\@gls@numlist@sep\@gls@numlist@lastsep
\@gls@thislist
\egroup
}%
}
}
\newcommand*\glsnumlistsep{, }
\newcommand*\glsnumlistlastsep{ \& }
\newcommand*\gls hyperlink}[2][\glsentrytext{\@glo@label}]{%
  \def\@glo@label{#2}%
  \@glslink{\glo linkprefix\glsdetoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*\glsadd}[2][{}]{%
  \@gls@adjustmode
  \glsdoifexists{#2}%
  {%
    \def\@glsnumberformat{\glsnumberformat}%
    \edef\@gls@counter{\csname glo@\glsdetoklabel{#2}@counter\endcsname}%
    \setkeys{glossadd}{#1}%
    \@gls@saveentrycounter
    \@gls@setsort{#2}%
    \@do@wrglossary{#2}%
  }%
}
\newcommand*\@gls@adjustmode{}
\AtBeginDocument{\renewcommand*\@gls@adjustmode{\ifvmode\mbox{}\fi}}

```

```

\newrobustcmd*\glsaddall}[1] [] {%
  \edef\glo@type{\@glo@types}%
  \setkeys{glossadd}{#1}%
  \forallglsentries[\@glo@type]{\@glo@entry}{%
    \glsadd[#1]{\@glo@entry}%
  }%
}
\newrobustcmd*\glsaddallunused}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glo@entry}%
  {%
    \ifglsused{\@glo@entry}{\glsadd[format=glsignore]{\@glo@entry}}%
  }%
}
\newcommand*\glsignore}[1]{}
\edef\glsopenbrace{\expandafter\@gobble\string\{ }
\edef\glsclosebrace{\expandafter\@gobble\string\} }
\edef\glsbackslash{\expandafter\@gobble\string\ }
\edef\glsquote#1{\string"#1\string"}
\edef\glspercentchar{\expandafter\@gobble\string\% }
\edef\glsstildechar{\string~ }
\ifglsxindy
  \newcommand*\@glsfirstletter}{A}
\fi
\newcommand*\GlsSetXdyFirstLetterAfterDigits){%
  \@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
\ifglsxindy
  \newcommand*\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*\@glsfirstletter}{#1}}
  \newcommand*\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*\@glsfirstletter}{#1}%
    \@onelevel@sanitize\@glsfirstletter
  }
\else
  \newcommand*\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \glsnoxywarning\GlsSetXdyFirstLetterAfterDigits}
  \newcommand*\s@GlsSetXdyFirstLetterAfterDigits){%
    \GlsSetXdyFirstLetterAfterDigits
  }
\fi
\ifglsxindy
  \newcommand*\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
\fi
\newcommand*\GlsSetXdyNumberGroupOrder}[1]{%
  \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
}
\ifglsxindy
  \newcommand*\@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*\@xdynumbergrouporder}{#1}%
  }
  \newcommand*\s@GlsSetXdyNumberGroupOrder}[1]{%

```

```

\renewcommand*\@xdynumbergrouporder}{#1}%
\@onelevel@sanitize\@xdynumbergrouporder
}
\else
\newcommand*\@GlsSetXdyNumberGroupOrder}[1]{%
\glsnoxywarning\GlsSetXdyNumberGroupOrder}
\newcommand*\s@GlsSetXdyNumberGroupOrder}{%
\@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*\@glsminrange}{2}
\ifglsxindy
\newcommand*\@GlsSetXdyMinRangeLength}[1]{%
\renewcommand*\@glsminrange}{#1}}
\else
\newcommand*\@GlsSetXdyMinRangeLength}[1]{%
\glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglsxindy
\def\writeist{%
\ifundef\glswrite{\newwrite\glswrite}{}%
\@gls@addpredefinedattributes
\openout\glswrite=\istfilename
\write\glswrite{;; xindy style file created by the glossaries
package}%
\write\glswrite{;; for document '\jobname' on
\the\year-\the\month-\the\day}%
\write\glswrite{^^J; required styles^^J}
\@for\@xdystyle:=\@xdyrequiredstyles\do{%
\ifx\@xdystyle\@empty
\else
\protected@write\glswrite}{(require
\string"\@xdystyle.xdy\string")}%
\fi
}%
\write\glswrite{^^J%
; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%
\@for\@gls@classI:=\@gls@xdy@locationlist\do{%
\protected@write\glswrite}{(define-location-class
\string"\@gls@classI\string"^^J\space\space\space
(
:sep "{ }{"
\csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
:sep "}"
)
^^J\space\space\space

```

```

: min-range-length \@glsminrange^^J%
)
}%
{%
\@for\@gls@classII:=\@gls@xdy@locationlist\do{%
\protected@write\glswrite{}\{(define-location-class
\string"\@gls@classII-\@gls@classI\string"
^^J\space\space\space
(
: sep "{"
\csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
: sep "}\{"
\csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
: sep "}"
)
^^J\space\space\space
: min-range-length \@glsminrange^^J%
)
}%
}%
}%
}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
:unverified )}%
\write\glswrite{(markup-crossref-list
: class \string"see\string"^^J\space\space\space
: open \string"\string\glsseeformat\string"
: close \string"{}\string")}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
(\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
: open \string"\string
\glossarysection[\string\glossarytoctitle]{\string
\glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
{%
\@for\@this@attr:=\@xdyattributelist\do{%
\protected@write\glswrite{}\{\string\providecommand*
\expandafter\string
\csname glsX\@this@ctr X\@this@attr\endcsname[2]%
{%
\string\setentrycounter
[\expandafter\@gobble\string\#1]{\@this@ctr}%
\expandafter\string

```

```

\csname \@this@attr\endcsname
  {\expandafter\@gobble\string\#2}%
}%
}%
}%
}%
}%
\write\glswrite{%
  \string\begin
  {theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
  \space\space:close \string" \glpercentchar\glstildechar n\string
  \end{theglossary}\string\glossarypostamble
  \glstildechar n\string" ^^J\space\space\space
  :tree)}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \glstildechar n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list
  :sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
  :sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
  :open-head \string"\string\glsgroupheading
  \glsopenbrace\string"^^J\space\space\space
  :close-head \string"\glsclosebrace\string")}%

```

```

\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}%
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax
\ifundef{\glswrite}{\newwrite\glswrite}{}\relax
\openout\glswrite=\istfilename
\write\glswrite{\glspercentchar\space makeindex style file
created by the glossaries package}
\write\glswrite{\glspercentchar\space for document
'\jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\@gls@actualchar'}
\write\glswrite{encap '\@gls@encapchar'}
\write\glswrite{level '\@gls@levelchar'}
\write\glswrite{quote '\@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
\glossarytoctitle]{\string\glossarytitle}\string
\glossarypreamble\string\n\string\begin{theglossary}\string
\glossaryheader\string\n\string"}
\write\glswrite{postamble \string"\string%\string\n\string
\end{theglossary}\string\glossarypostamble\string\n
\string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
\string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}
\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{delim_0 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string\relax \string"}
\glossaryentrynumbers\string\{\string\relax \string"}

```

```

\write\glswrite{delim_2 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string}
\write\glswrite{delim_t \string"\string}\string}\string"}
\write\glswrite{delim_n \string"\string\delimN \string"}
\write\glswrite{delim_r \string"\string\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
  \string"\string}\string\relax
  \string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"glscpositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\fi
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{
\ifglxsindy
\newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\newcommand*\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}%
  \@ifpackageloaded{tracklang}%
  {%
    \IfTrackedLanguage{german}%
    {%
      \def\@gls@extramakeindexopts{-g}%
    }%
  }%
}
\edef\@gls@docheckquotedef{%
  \noexpand\def\noexpand\@gls@checkquote####1#1####2#1####3\noexpand\null{%
    \noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
    \noexpand\toks@={###1}%

```

```

\noexpand\ifx\noexpand\null###2\noexpand\null
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###2#1#1\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###2#1###3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkquote###1\noexpand\@nil
#1#1\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkesquote###1\noexpand\@nil
\expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkesactual###1\noexpand\@nil
\noexpand\?\noexpand\?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkactual###1\noexpand\@nil
\noexpand?\noexpand?\noexpand\null

```



```

\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkbar###1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkesobar###1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checklevel###1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkesquote###1%
\expandonce{\csname#1\endcsname}###2\expandonce{\csname#1\endcsname}%
###3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={###1}%
\noexpand\ifx\noexpand\null###2\noexpand\null
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@@gls@checkesquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@@gls@checkesquote{%
\noexpand\@gls@checkesquote###3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string
\expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@@gls@checkesquote{%
\noexpand\@gls@checkesquote###2\expandonce{\csname#1\endcsname}%
\expandonce{\csname#1\endcsname}\noexpand\null}%

```

```

        \noexpand\else
        \noexpand\def\noexpand\@gls@checkescquote{%
            \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
            ####3\noexpand\null}%
        \noexpand\fi
        \noexpand\fi
        \noexpand\@gls@checkescquote
    }%
}
\@gls@docheckquotedef
}
\newcommand*\@gls@nosetquote}[1]{\PackageError{glossaries}%
    {\string\GlsSetQuote\space not permitted here}%
    {Move \string\GlsSetQuote\space earlier in the preamble, as
    soon as possible after glossaries.sty has been loaded}}
\fi
\newcommand*\@gls@extramakeindexopts}[1]{
\newcommand*\@noist}{%
    \@gls@addpredefinedattributes
    \let\writeist\relax
}
\newcommand*\@makeglossary}[1]{%
    \ifglossaryexists{#1}%
    {%
        \ifglssavewrites
            \expandafter\newtoks\csname glo@#1@filetok\endcsname
        \else
            \expandafter\newwrite\csname glo@#1@file\endcsname
            \expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
        \fi
        \@gls@renewglossary
        \writeist
    }%
    {%
        \PackageError{glossaries}%
        {Glossary type ‘#1’ not defined}%
        {New glossaries must be defined before using \string\makeglossaries}%
    }%
}
\newcommand*\@glsopenfile}[2]{%
    \immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
    \PackageInfo{glossaries}{Writing glossary file
        \jobname.\csname @glotype@#2@out\endcsname}%
}
\newcommand*\@closegls}[1]{%
    \closeout\csname glo@#1@file\endcsname
}
\ifglsexindy
\newcommand*\@gls@automake}[1]{%
    \ifglossaryexists{#1}

```

```

{%
  \@closegls{#1}%
  \ifdefstring{\glsorder}{letter}%
  {\def\@gls@order{-M ord/letorder }}%
  {\let\@gls@order\@empty}%
  \ifcsundef{@xdy@#1@language}%
  {\let\@gls@langmod\@xdy@main@language}%
  {\letcs\@gls@langmod{@xdy@#1@language}}%
  \edef\@gls@dothiswrite{\noexpand\write18{xindy
    -I xindy
    \@gls@order
    -L \@gls@langmod\space
    -M \@gls@istfilebase\space
    -C \@gls@codepage\space
    -t \jobname.\csuse{@glotype@#1@log}
    -o \jobname.\csuse{@glotype@#1@in}
    \jobname.\csuse{@glotype@#1@out}}}%
  }%
  \@gls@dothiswrite
}%
{%
  \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\else
\newcommand*{\@gls@automake}[1]{%
  \ifglossaryexists{#1}
  {%
    \@closegls{#1}%
    \ifdefstring{\glsorder}{letter}%
    {\def\@gls@order{-l }}%
    {\let\@gls@order\@empty}%
    \edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
      -s \istfilename\space
      -t \jobname.\csuse{@glotype@#1@log}
      -o \jobname.\csuse{@glotype@#1@in}
      \jobname.\csuse{@glotype@#1@out}}}%
    }%
    \@gls@dothiswrite
  }%
  {%
    \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
  }%
}
\fi
\ifglsxindy
\newcommand*{\@gls@automake@immediate}[1]{%
  \ifglossaryexists{#1}
  {%
    \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%

```

```

{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-M ord/letorder }}%
{\let\@gls@order\@empty}%
\ifcsundef{@xdy@#1@language}%
{\let\@gls@langmod\@xdy@main@language}%
{\letcs\@gls@langmod{@xdy@#1@language}}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \gls@istfilename\space
-C \gls@codepage\space
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{\GlossariesWarning{Can't make glossary '#1', it doesn't exist}}%
}%
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%
\IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-l }}%
{\let\@gls@order\@empty}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order
-s \istfilename\space
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{\GlossariesWarning{Can't make glossary '#1', it doesn't exist}}%
}%
}
}

```

```

\fi
\newcommand*{\@warn@nomakeglossaries}{}
\newcommand*{\warn@nomakeglossaries}{\@warn@nomakeglossaries}
\newcommand{\@gls@automake@immediate}{%
\ifnum\gls@automake@nr=2\relax
\@for\@gls@type:=\@glo@types\do{%
\ifdefempty{\@gls@type}{}%
{\@gls@automake@immediate{\@gls@type}}%
}%
\glsautomakefalse
\renewcommand*{\@gls@doautomake}{}%
\fi
}
\newcommand*{\makeglossaries}{%
\@domakeglossaries
{%
\@gls@@automake@immediate
\ifundef{\glswrite}{\newwrite\glswrite}{}%
\protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{} }
\protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{} }
\ifundef\@@gls@extramakeindexopts
{}%
{%
\protected@write\@auxout{}{\string\providecommand
\string\@gls@extramakeindexopts[1]{} }
\protected@write\@auxout{}{\string\@gls@extramakeindexopts
{\@@gls@extramakeindexopts}}%
}%
\protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
\protected@write\@auxout{}{\string\@glsorder{\glsorder}}
\@for\@glo@type:=\@glo@types\do{%
\ifthenelse{\equal{\@glo@type}{} }{}{}%
\@makeglossary{\@glo@type}}%
}%
\renewcommand*\newglossary[4] []{%
\PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%
\let\@makeglossary\@gobble
\let\makeglossaries\relax
\@disable@onlypremakeg
\let\gls@checkseeallowed\relax
\let\warn@nomakeglossaries\relax
\def\warn@noprintglossary{%
\ifdefstring{\@glo@types}{,}%
{%
\GlossariesWarningNoLine{No glossaries have been defined}%
}%
}%

```

```

\GlossariesWarningNoLine{No \string\printglossary\space
or \string\printglossaries\space
found. ^^J(Remove \string\makeglossaries\space if you
don't want any glossaries.) ^^JThis document will not
have a glossary}%
}%
}%
\ifglssavenumberlist
\edef\@gls@dodolistparser{\noexpand\DeclareListParser
{\noexpand\glsnumlistparser}{\delimN}}%
\@gls@dodolistparser
\fi
\let\makenoidxglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
\let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
\ifglsautomake
\renewcommand*{\@gls@doautomake}{%
\for\@gls@type:=\@glo@types\do{%
\ifdefempty{\@gls@type}{%
{\@gls@automake{\@gls@type}}%
}%
}%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}
\AtEndDocument{%
\warn@nomakeglossaries
\warn@noprintglossary
}
\newcommand*{\makenoidxglossaries}{%
\@domakeglossaries
{%
\renewcommand{\@gls@nofwarn}[1]{%
\GlossariesWarning{Empty glossary for
\string\printnoidxglossary[type={##1}].
Rerun may be required (or you may have forgotten to use
commands like \string\gls)}%
}%
\let\@gls@checkmkidxchars\@gobble
\glsesclocationsfalse
\let\@do@wrglossary\gls@noidxglossary
\let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
\let\gls@checkseeallowed\relax
\renewcommand{\@do@seeglossary}[2]{%
\edef\@gls@label{\glsdetoklabel{##1}}%
\protected@write\@auxout{}{%
\string\@gls@reference

```

```

        {\csname glo@\@gls@label @type\endcsname}%
        {\@gls@label}%
        {%
        \string\glsseeformat##2}%
    }%
}%
}%
\AtBeginDocument
{%
\write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
}%
\def\warn@noprintglossary{%
\GlossariesWarningNoLine{No \string\printnoidxglossary\space
or \string\printnoidxglossaries ^^J
found. (Remove \string\makenoidxglossaries\space if you
don't want any glossaries.)^^JThis document will not have a glossary}%
}%
\let\warn@nomakeglossaries\relax
\let\makeglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
\let\@glo@assign@sortkey\@glo@assign@sortkey
\def\@glo@sorttype{\@glo@default@sorttype}%
}%
\renewcommand*\new@glossaryentry[2]{%
\PackageError{glossaries}{Glossary entries must be
defined in the preamble^^Jwhen you use
\string\makenoidxglossaries}%
{Either move your definitions to the preamble or use
\string\makeglossaries}%
}%
\renewcommand*\@gls@entrynumberlist[1]{%
\letcs{\@gls@loclist}{glo@\@gls@detoklabel{##1}@loclist}%
\ifdef\@gls@loclist
{%
\glsnoidxloclist{\@gls@loclist}%
}%
}%
{%
??\glsdoifexists{##1}%
}%
\GlossariesWarning{Missing location list for ‘##1’. Either
a rerun is required or you haven't referenced the entry}%
}%
}%
}%
\renewcommand*\@gls@displaynumberlist[1]{%
\letcs{\@gls@loclist}{glo@\@gls@detoklabel{##1}@loclist}%
\ifdef\@gls@loclist
{%
\def\@gls@noidxloclist@sep{%
\def\@gls@noidxloclist@sep{%

```

```

        \def\@gls@noidxloclist@sep{%
            \glsnumlistsep
        }%
        \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
    }%
}%
\def\@gls@noidxloclist@finalsep{}%
\def\@gls@noidxloclist@prev{}%
\forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
\@gls@noidxloclist@finalsep
\@gls@noidxloclist@prev
}%
{%
    ??\glsdoifexists{##1}%
    {%
        \GlossariesWarning{Missing location list for ‘##1’. Either
            a rerun is required or you haven’t referenced the entry}%
    }%
}%
}%
\renewcommand*{\glsnumberlistloop}[3]{%
    \letcs{\@gls@loclist}{glo\@glsdetoklabel{##1}@loclist}%
    \let\@gls@org@glsnoidxdisplayloc@glsnoidxdisplayloc
    \let\@gls@org@glsseeformat@glsseeformat
    \let\glsnoidxdisplayloc##2\relax
    \let\glsseeformat##3\relax
    \ifdef\@gls@loclist
    {%
        \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
    }%
    {%
        ??\glsdoifexists{##1}%
        {%
            \GlossariesWarning{Missing location list for ‘##1’. Either
                a rerun is required or you haven’t referenced the entry}%
        }%
    }%
    \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
    \let\glsseeformat\@gls@org@glsseeformat
}%
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
\@glo@check@sortallowed\makenoidxglossaries
}%
}
\@onlypreamble{\makenoidxglossaries}
\newcommand*{\glsnumberlistloop}[2]{%
    \PackageError{glossaries}{\string\glsnumberlistloop\space
        only works with \string\makenoidxglossaries}{}%
}

```



```

}
\newcommand*\glsnoidxnumberlistloophandler}[1]{%
  #1%
}
\newcommand*\@no@makeglossaries{%
  \PackageError{glossaries}{You can't use both
  \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
  together.}%
}
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
  is required to make \string\printnoidxglossary[type={#1}] work}%
}
\newcommand*\@gls@noidxglossary{%
  \protected@write\@auxout{%
    \string\@gls@reference
    {\csname glo@\@gls@label @type\endcsname}%
    {\@gls@label}%
    {\string\glsnoidxdisplayloc
    {\@glo@counterprefix}%
    {\@gls@counter}%
    {\@glsnumberformat}%
    {\@glslocref}%
    }%
  }%
}
}
\providecommand\istfile{\glswrite}
\AtEndDocument{%
  \glswritefiles
}
\newcommand*\@gls@writefiles{%
  \forallglossaries{\@glo@type}{%
    \ifcsundef{glo@\@glo@type @filetok}%
    {%
      \def\gls@tmp{%
    }%
    }%
    {\def\gls@tmp{\expandafter\the
      \csname glo@\@glo@type @filetok\endcsname}%
    }%
    \ifx\gls@tmp\@empty
      \ifx\@glo@type\glsdefaulttype
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries.^^JRemember to use package option 'nomain' if
you
          don't want to^^Juse the main glossary}%
      \else
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries}%
      \fi
    }%
  }%
}

```

```

        \fi
    \else
        \@glsopenfile{\glswrite}{\@glo@type}%
        \immediate\write\glswrite{%
            \expandafter\the
            \csname glo@\@glo@type @filetok\endcsname}%
        \immediate\closeout\glswrite
    \fi
}%
}
\newcommand*\@gls@glossary}[1]{%
    \@gls@glossary{#1}%
}
\newcommand*\@gls@glossary}[2]{%
    \if@gls@debug
        \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
    \fi
}
\newcommand*\@gls@renewglossary{%
    \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
    \let\@gls@renewglossary\@empty
}
\newcommand*\@gls@wrglossary}[2]{%
    \ifgls@savewrites
        \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
        \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
            \expandafter{\@gls@tmp~J}%
    \else
        \ifcsdef{glo@#1@file}%
            {%
                \expandafter\protected@write\csname glo@#1@file\endcsname{%
                    \gls@disablepagerefexpansion}{#2}%
            }%
            {%
                \ifignoredglossary{#1}{}%
                    {%
                        \GlossariesWarning{No file defined for glossary ‘#1’}%
                    }%
            }%
        \fi
    \endgroup\@esphack
}
\newcommand*\@do@wrglossary}[1]{%
    \glswriteentry{#1}{\@do@wrglossary{#1}}%
}
\newcommand*\@gls@writeentry}[2]{%
    \ifgls@indexonlyfirst
        \ifglsused{#1}{#2}%
    \else
        #2%
    \fi
}

```

```

\fi
}
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
\newcommand*{\gls@disablepagerefexpansion}{%
\@for\@gls@this:=\gls@protected@pagefmts\do
{%
\expandafter\let\@gls@this\relax
}%
}
\newcommand*{\gls@alphpage}{\@alph\c@page}
\newcommand*{\gls@Alphpage}{\@Alph\c@page}
\newcommand*{\gls@numberpage}{\number\c@page}
\newcommand*{\gls@arabicpage}{\@arabic\c@page}
\newcommand*{\gls@romanpage}{\romannumeral\c@page}
\newcommand*{\gls@Romanpage}{\@Roman\c@page}
\newcommand*{\glsaddprotectedpagefmt}[1]{%
\ea\ptto\gls@protected@pagefmts{\expandonce{\csname gls#1page\endcsname}}%
\csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
\ea\ptto\@wrglossarynumberhook{%
\noexpand\let\expandonce{\csname org@gls#1\endcsname}%
\expandonce{\csname#1\endcsname}%
\noexpand\def\expandonce{\csname#1\endcsname}{%
\noexpand\@wrglossary@pageformat
\expandonce{\csname gls#1page\endcsname}%
\expandonce{\csname org@gls#1\endcsname}%
}%
}%
}
\newcommand*\@wrglossarynumberhook{}
\newcommand{\@wrglossary@pageformat}[3]{%
\ifx#3\c@page #1\else #2#3\fi
}
\newcommand*{\@do@wrglossary}[1]{%
\ifglseclocations
\@do@esc@wrglossary{#1}%
\else
\@do@noesc@wrglossary{#1}%
\fi
}
\newcommand*{\@do@noesc@wrglossary}[1]{%
\expandafter\def\expandafter\@glslocref\expandafter{\theglentrycounter}%
\expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
\ifx\@glsHlocref\@glslocref
\def\@glo@counterprefix{}%
\else
\protected@edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
{\@glslocref}{\@glsHlocref}%
}%
\@do@gls@getcounterprefix

```

```

\fi
\edef\@gls@label{\glsdetoklabel{#1}}%
\@do@wrglossary
}
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodstrue
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
\begingroup
\let\gls@orgthe\the
\let\gls@orgnumber\number
\let\gls@orgarabic\@arabic
\let\gls@orgromannumeral\romannumeral
\let\gls@orgalph\@alph
\let\gls@orgAlph\@Alph
\let\gls@orgRoman\@Roman
\ifglswrallowprimitivemods
\def\gls@the##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
\def\the{\expandafter\gls@the}%
\def\gls@number##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
\def\number{\expandafter\gls@number}%
\fi
\def\@arabic##1{%
\ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\romannumeral##1{%
\ifx##1\c@page \gls@romanpage\else\gls@orgromannumeral##1\fi}%
\def\@Roman##1{%
\ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
\ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
\ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\@gls@locref{\thegl@sentrycounter}%
\endgroup
\@gls@checkmkidxchars\@gls@locref
\expandafter\ifx\theHgl@sentrycounter\thegl@sentrycounter\relax
\def\@glo@counterprefix{%
\else
\protected@edef\@gls@Hlocref{\theHgl@sentrycounter}%
\@gls@checkmkidxchars\@gls@Hlocref
\edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
{\@gls@locref}{\@gls@Hlocref}%
}%
\@do@gls@getcounterprefix
\fi
\edef\@gls@label{\glsdetoklabel{#1}}%
\@do@wrglossary

```

```

}
\newcommand*{\@do@wrglossary}{%
  \ifglxindy
    \expandafter\@glo@check@mkidxrangear\@glsnumberformat\@nil
    \def\@glo@range{}%
    \expandafter\if\@glo@prefix(\relax
      \def\@glo@range{:open-range}%
    \else
      \expandafter\if\@glo@prefix)\relax
      \def\@glo@range{:close-range}%
    \fi
  \fi
  \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
    (indexentry :tkey (\csname glo@\@gls@label @index\endcsname)
      :locref \string{\@glo@counterprefix}{\@glslocref}\string" %
      :attr \string"\@gls@counter\@glo@suffix\string"
      \@glo@range
    )
  }%
\else
  \set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@glsnumberformat}%
  {\@glo@counterprefix}%
  \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
    \string@glossaryentry{\csname glo@\@gls@label @index\endcsname
      \@gls@encapchar\@glo@numfmt}{\@glslocref}}%
  \fi
}
\newcommand*\@gls@getcounterprefix[2]{%
  \edef\@gls@thisloc{#1}\edef\@gls@thisHloc{#2}%
  \ifx\@gls@thisloc\@gls@thisHloc
    \def\@glo@counterprefix{}%
  \else
    \def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
      \def\@glo@tmp{##2}%
      \ifx\@glo@tmp\@empty
        \def\@glo@counterprefix{}%
      \else
        \def\@glo@counterprefix{##1}%
      \fi
    }%
    \@gls@get@counterprefix#2.#1\end@getprefix
  \ifx\@glo@counterprefix\@empty
    \GlossariesWarning{Hyper target '#2' can't be formed by
      prefixing location '#1'. You need to modify the
      definition of \string\theH\@gls@counter\@Jotherwise you
      will get the warning: "'name{\@gls@counter.#1}' has been
      referenced but does not exist"}%
  \fi
\fi
}

```

```

\newcommand{\@do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref
\ifglsxindy
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    (indexentry
      :tkey (\csname glo@#1@index\endcsname)
      :xref (\string"\@gls@xref\string")
      :attr \string"see\string"
    )
  }%
\else
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    \string\glossaryentry{\csname glo@#1@index\endcsname
    \@gls@encapchar glsseeformat\@gls@xref}{Z}}%
\fi
}
\def\@gls@fixbraces#1#2#3\@nil{%
  \ifx#2[\relax
    \@gls@fixbraces#1#2#3\@end@fixbraces
  \else
    \def#1{#{#2#3}}%
  \fi
}
\def\@gls@fixbraces#1[#2]#3\@end@fixbraces{%
  \def#1{[#2]{#3}}%
}
\newrobustcmd*\@glssee}[3][\seename]{%
  \@do@seeglossary{#2}{[#1]{#3}}
\newcommand*\@glssee}[3][\seename]{%
  \glssee[#1]{#3}{#2}}
\newrobustcmd*\@glsseeformat}[3][\seename]{%
  \emph{#1} \glsseeitem{#2}}
\newrobustcmd*\@glsseeelist}[1]{%
  \let\@gls@dolast\relax
  \let\@gls@donext\relax
  \@for\@gls@thislabel:=#1\do{%
    \ifx\@xfor@nextelement\@nnil
      \@gls@dolast
    \else
      \@gls@donext
    \fi
    \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
    \let\@gls@dolast\glsseeitem
    \let\@gls@donext\glsseeitem
  }%
}
\newcommand*\@glsseeitem{\space\andname\space}
\newcommand*\@glsseeitemsep{, }

```

```

\newrobustcmd*{\glsseeitem}[1]{\glshyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*{\glsseeitemformat}[1]{\glsentrytext{#1}}
\newcommand*{\gls@save@numberlist}[1]{%
  \ifgls@save@numberlist
    \toks@{#1}%
    \edef\@do@writeaux@info{%
      \noexpand\csgdef{glo@\gls@current@entry@label @numberlist}{\the\toks@}%
    }%
    \@onelevel@sanitize\@do@writeaux@info
    \protected@write\@auxout{}{\@do@writeaux@info}%
  \fi
}
\newcommand*{\warn@noprntglossary}{}%
\ifcsundef{printglossary}{}%
{
  \@gls@warnonglossdefined
  \undef\printglossary
}
\newcommand*{\@printgloss@checkexists@noignored}[2]{%
  \ifglossaryexists{#1}%
  {#2}%
  {
    \ifignoredglossary{#1}%
    {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
    {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
  }%
}
\newcommand*{\@printgloss@checkexists@allowignored}[2]{%
  \s@ifglossaryexists{#1}%
  {#2}%
  {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\newcommand*{\printglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@glossary}%
}
\newcommand*{\printglossaries}{%
  \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}
\newcommand*{\printnoidxglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@noidx@glossary}%
}
\newcommand*{\printnoidxglossaries}{%
  \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
\newcommand*{\@printgloss@setsort}{%
}
\newcommand*{\@gls@preglossaryhook}{%
}
\newcommand{\@printglossary}[2]{%

```

```

\def@glo@type{glsdefaulttype}%
\def@glossarytitle{\csname @glo@type @title\endcsname}%
\def@glossarytoctitle{glossarytitle}%
\let\org@glossarytitle@glossarytitle
\def@glossarystyle{%
  \ifx@glossary@default@style\relax
    \GlossariesWarning{No default glossary style provided \MessageBreak
      for the glossary '@glo@type'. \MessageBreak
      Using deprecated fallback. \MessageBreak
      To fix this set the style with \MessageBreak
      \string\setglossarystyle\space or use the \MessageBreak
      style key=value option}%
  \fi
}%
\def@gls@dotoc@title{glssettoctitle{@glo@type}}%
\let@org@glossaryentrynumbers@glossaryentrynumbers
\bgroup
  \@printgloss@setsort
  \setkeys{printgloss}{#1}%
  \@printgloss@checkexists{@glo@type}%
  {%
    \ifx@glossarytitle\org@glossarytitle
    \else
      \expandafter\let\csname @glo@type @title\endcsname
        \glossarytitle
    \fi
    \let\currentglossary@glo@type
    \let\org@glossaryentrynumbers@glossaryentrynumbers
    \let\glsnonextpages@glsnonextpages
    \let\glsnextpages@glsnextpages
    \let\nopostdesc@nopostdesc
    \gls@dotoc@title
    \@glossarystyle
    \let\gls@org@glossaryentryfield@glossentry
    \let\gls@org@glossarysubentryfield\subglossentry
    \renewcommand{\glossentry}[1]{%
      \xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
      \gls@org@glossaryentryfield{##1}%
    }%
    \renewcommand{\subglossentry}[2]{%
      \xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
      \gls@org@glossarysubentryfield{##1}{##2}%
    }%
    \@gls@preglossaryhook
    #2%
  }%
\egroup
\global\let@glossaryentrynumbers@org@glossaryentrynumbers
\global\let\warn@noprintglossary\relax
}

```



```

\newcommand{\@print@glossary}{%
  \makeatletter
  \@input@{\jobname.\csname @glo@type@\glo@type @in\endcsname}%
  \IfFileExists{\jobname.\csname @glo@type@\glo@type @in\endcsname}%
  {}%
  {\null}%
  \ifglxindy
    \ifcsundef{@xdy@\glo@type @language}%
    {%
      \edef\@do@auxoutstuff{%
        \noexpand\AtEndDocument{%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\providecommand\string\@xdylanguage[2]{}%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\@xdylanguage{\glo@type}{\@xdy@main@language}%
          }%
        }%
      }%
    }%
    {%
      \edef\@do@auxoutstuff{%
        \noexpand\AtEndDocument{%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\providecommand\string\@xdylanguage[2]{}%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\@xdylanguage{\glo@type}{\csname @xdy@\glo@type
              @language\endcsname}%
          }%
        }%
      }%
    }%
  \do@auxoutstuff
  \edef\@do@auxoutstuff{%
    \noexpand\AtEndDocument{%
      \noexpand\immediate\noexpand\write\@auxout{%
        \string\providecommand\string\@gls@codepage[2]{}%
      \noexpand\immediate\noexpand\write\@auxout{%
        \string\@gls@codepage{\glo@type}{\gls@codepage}%
      }%
    }%
  }%
  \do@auxoutstuff
\fi
\renewcommand*{\@warn@nomakeglossaries}{%
  \GlossariesWarningNoLine{\string\makeglossaries\space
  hasn't been used,^^Jthe glossaries will not be updated}%
}%
}
\newcommand*{\@glo@sortentries}[2]{%
  \glosortentrieswarning
  \def\@glo@sortinglist{}%
  \def\@glo@sortinghandler{#1}%
  \edef\@glo@type{#2}%

```

```

\forlistcsloop{\@glo@do@sortentries}{@glsref@#2}%
\csdef{@glsref@#2}{}%
\@for\@this@label:=\@glo@sortinglist\do{%
  \xifinlistcs{\@this@label}{@glsref@#2}%
  {%
  {%
    \listcsxadd{@glsref@#2}{\@this@label}%
  }%
  \ifcsdef{@glo@sortingchildren@\@this@label}%
  {%
    \@glo@addchildren{#2}{\@this@label}%
  }%
  }%
}%
}
\newcommand*{\@glo@addchildren}[2]{%
  \bgroup
  \letcs{\@glo@childlist}{@glo@sortingchildren@#2}%
  \@for\@this@childlabel:=\@glo@childlist\do
  {%
    \xifinlistcs{\@this@childlabel}{@glsref@#1}%
    {%
    {%
      \listcsxadd{@glsref@#1}{\@this@childlabel}%
    }%
    \ifcsdef{@glo@sortingchildren@\@this@childlabel}%
    {%
      \@glo@addchildren{#1}{\@this@childlabel}%
    }%
    }%
  }%
  }%
  \egroup
}
\newcommand*{\@glo@do@sortentries}[1]{%
  \ifglshasparent{#1}%
  {%
  \edef\@glo@parent{\csuse{glo@\glsdetoklabel{#1}@parent}}%
  \ifcsundef{@glo@sortingchildren@\@glo@parent}%
  {%
    \csdef{@glo@sortingchildren@\@glo@parent}{}%
  }%
  }%
  \expandafter\@glo@sortedinsert
  \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
  \xifinlistcs{\@glo@parent}{@glsref@\@glo@type}%
  {%
  }%
  {%
    \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
  }%
}

```

```

    }%
  }%
  {%
    \@glo@sortedinsert{\@glo@sortinglist}{#1}%
  }%
}
\newcommand*{\@glo@sortedinsert}[2]{%
  \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
}%
\newcommand*{\@glo@sorthandler@word}[2]{%
  \letcs\@gls@sort@A{glo@glsetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@glsetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@letter}[2]{%
  \letcs\@gls@sort@A{glo@glsetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@glsetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@case}[2]{%
  \letcs\@gls@sort@A{glo@glsetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@glsetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@nocase}[2]{%
  \letcs\@gls@sort@A{glo@glsetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@glsetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlicompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sortmacro@word}[1]{%

```

```

\ifdefstring{\@glo@default@sorttype}{standard}%
{%
  \@glo@sortentries{\@glo@sorthandler@word}{#1}%
}%
{%
  \PackageError{glossaries}{Conflicting sort options:^^J
  \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
  \string\printnoidxglossary[sort=word]}{}%
}%
}
\newcommand*\@glo@sortmacro@letter}[1]{%
\ifdefstring{\@glo@default@sorttype}{standard}%
{%
  \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
}%
{%
  \PackageError{glossaries}{Conflicting sort options:^^J
  \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
  \string\printnoidxglossary[sort=letter]}{}%
}%
}
\newcommand*\@glo@sortmacro@standard}[1]{%
\ifdefstring{\@glo@default@sorttype}{standard}%
{%
  \ifcsdef{\@glo@sorthandler@\glsorder}%
  {%
    \@glo@sortentries{\csuse{\@glo@sorthandler@\glsorder}}{#1}%
  }%
  {%
    \PackageError{glossaries}{Unknown sort handler ‘\glsorder’}{}%
  }%
}%
{%
  \PackageError{glossaries}{Conflicting sort options:^^J
  \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
  \string\printnoidxglossary[sort=standard]}{}%
}%
}
\newcommand*\@glo@sortmacro@case}[1]{%
\ifdefstring{\@glo@default@sorttype}{standard}%
{%
  \@glo@sortentries{\@glo@sorthandler@case}{#1}%
}%
{%
  \PackageError{glossaries}{Conflicting sort options:^^J
  \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
  \string\printnoidxglossary[sort=case]}{}%
}%
}
\newcommand*\@glo@sortmacro@nocase}[1]{%

```

```

\ifdefstring{\@glo@default@sorttype}{standard}%
{%
  \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
}%
{%
  \PackageError{glossaries}{Conflicting sort options:^^J
  \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
  \string\printnoidxglossary[sort=nocase]}{}%
}%
}
\newcommand*{\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{%
  \forglseentries[#1]{\@gls@thislabel}%
  {%
    \xifinlistcs{\@gls@thislabel}{\@glsref@#1}%
    {%
      \listeadadd{\@glo@sortinglist}{\@gls@thislabel}%
    }%
    {%
    }%
  }%
  \cslet{\@glsref@#1}{\@glo@sortinglist}%
}
\newcommand*{\@glo@sortmacro@def@do}[1]{%
  \ifinlistcs{#1}{\@glsref@\@glo@type}%
  {}%
  {%
    \listcsadd{\@glsref@\@glo@type}{#1}%
  }%
  \ifcsdef{\@glo@sortingchildren@#1}%
  {%
    \@glo@addchildren{\@glo@type}{#1}%
  }%
  {}%
}
\newcommand*{\@glo@sortmacro@use}[1]{%
\newcommand*{\@print@noidx@glossary}{%
  \ifcsdef{\@glsref@\@glo@type}%
  {%
    \ifcsdef{\@glo@sortmacro@\@glo@sorttype}%
    {%
      \csuse{\@glo@sortmacro@\@glo@sorttype}{\@glo@type}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}%
    }%
  }%
  \glossarysection[\@glossarytoctitle]{\@glossarytitle}%
  \glossarypreamble
  \def\@gls@currentlettergroup{}%
  \begin{theglossary}%

```

```

    \glossaryheader
    \glsresetentrylist
    \forlistcsloop{\@gls@noidx@do}{\@glsref@{\@glo@type}}%
    \end{theglossary}%
    \glossarypostamble
  }%
  {%
    \@gls@noref@warn{\@glo@type}%
  }%
}
\def\glo@grabfirst#1#2\@nil{%
  \def\@gls@firsttok{#1}%
  \ifdefempty\@gls@firsttok
  {%
    \def\@glo@thislettergrp{0}%
  }%
  {%
    \@onelevel@sanitize\@gls@firsttok
    \expandafter\@glo@grabfirst\@gls@firsttok{}}\@nil
  }%
}
\def\@glo@grabfirst#1#2\@nil{%
  \ifdefempty\@glo@thislettergrp
  {%
    \def\@glo@thislettergrp{glssymbols}%
  }%
  {%
    \count@=\uccode‘#1\relax
    \ifnum\count@=0\relax
      \def\@glo@thislettergrp{glssymbols}%
    \else
      \ifdefstring\@glo@sorttype{case}%
      {%
        \count@=‘#1\relax
      }%
      {%
      }%
    }%
    \edef\@glo@thislettergrp{\the\count@}%
  \fi
  }%
}
}
\newcommand{\@gls@noidx@do}[1]{%
  \global\letcs{\@gls@loclist}{glo@glstdetoklabel{#1}@loclist}%
  \ifglshasparent{#1}%
  {%
    \gls@level=\csuse{glo@glstdetoklabel{#1}@level}\relax
    \ifdefvoid{\@gls@loclist}
    {%
      \subglossentry{\gls@level}{#1}{}%
    }%
  }%
}

```

```

    {%
      \subglossentry{\gls@level}{#1}%
      {%
        \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
      }%
    }%
  }%
  {%
    \letcs{\@gls@sort}{glo@\glsdetoklabel{#1}@sort}%
    \expandafter\glo@grabfirst\@gls@sort{}{}\@nil
    \ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
    {%
      \ifdefempty{\@gls@currentlettergroup}{}%
      {%
        \global\let\@glo@thislettergrp\@glo@thislettergrp
        \glsgroupskip
      }%
      \glsgroupheading{\@glo@thislettergrp}%
    }%
    \global\let\@gls@currentlettergroup\@glo@thislettergrp
    \ifdefvoid{\@gls@loclist}
    {%
      \glossentry{#1}{}%
    }%
    {%
      \glossentry{#1}%
      {%
        \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
      }%
    }%
  }%
}
}
\newcommand*{\glsnoidxloclist}[1]{%
  \def\@gls@noidxloclist@sep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxloclisthandler}{#1}%
}
\newcommand*{\glsnoidxloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%
    \@gls@noidxloclist@sep
    #1%
    \def\@gls@noidxloclist@sep{\delimN}%
    \def\@gls@noidxloclist@prev{#1}%
  }%
}
\newcommand*{\glsnoidxdisplayloclisthandler}[1]{%

```

```

\ifdefstring{\@gls@noidxloclist@prev}{#1}%
{%
}%
{%
  \@gls@noidxloclist@sep
  \@gls@noidxloclist@prev
  \def\@gls@noidxloclist@prev{#1}%
}%
}
\newcommand*\glsnoidxdisplayloc[4]{%
  \setentrycounter[#1]{#2}%
  \csuse{#3}{#4}%
}
\newcommand*\@gls@reference}[3]{%
  \glsdoifexistsorwarn{#2}%
  {%
    \ifcsundef{\@glsref@#1}{\csgdef{\@glsref@#1}{}}{}%
    \ifinlistcs{#2}{\@glsref@#1}%
    {}%
    {\listcsgadd{\@glsref@#1}{#2}}%
    \ifcsundef{glo@\glsdetoklabel{#2}@loclist}%
    {\csgdef{glo@\glsdetoklabel{#2}@loclist}{}}%
    {}%
    \listcsgadd{glo@\glsdetoklabel{#2}@loclist}{#3}%
  }%
}
\define@key{printgloss}{type}{\def\@glo@type{#1}}
\define@key{printgloss}{title}{%
  \def\glossarytitle{#1}%
  \let\gls@dotocitle\relax
}
\define@key{printgloss}{toctitle}{%
  \def\glossarytoctitle{#1}%
  \let\gls@dotocitle\relax
}
\define@key{printgloss}{style}{%
  \ifcsundef{\@glsstyle@#1}%
  {%
    \PackageError{glossaries}%
    {Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \def\@glossarystyle{\setglossentrycompatibility
      \csname @glsstyle@#1\endcsname}%
  }%
}
\define@choicekey{printgloss}{numberedsection}%
[\@gls@numberedsection@val\@gls@numberedsection@nr]%
{false,nolabel,autolabel,nameref}[nolabel]%
{}

```



```

\ifcase\gls@numberedsection@nr\relax
  \renewcommand*{\@@glossarysecstar}{*}%
  \renewcommand*{\@@glossaryseclabel}{}%
\or
  \renewcommand*{\@@glossarysecstar}{}%
  \renewcommand*{\@@glossaryseclabel}{}%
\or
  \renewcommand*{\@@glossarysecstar}{}%
  \renewcommand*{\@@glossaryseclabel}{\label{\glsautoprefix\@glo@type}}%
\or
  \renewcommand*{\@@glossarysecstar}{*}%
  \renewcommand*{\@@glossaryseclabel}{%
    \protected@edef\@currentlabelname{\glossarytoctitle}%
    \label{\glsautoprefix\@glo@type}}%
\fi
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
  \csuse{glsnogroupskip#1}%
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
  \csuse{glsnopostdot#1}%
}
\newcommand*{\GlsEntryCounterLabelPrefix}{glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
  \csuse{glsentrycounter#1}%
  \@gls@define@glossaryentrycounter
}
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
  \csuse{glsesubentrycounter#1}%
  \@gls@define@glossarysubentrycounter
}
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
\ifglsnonumberlist
  \def\glossaryentrynumbers##1{%
\else
  \def\glossaryentrynumbers##1{##1}%
\fi}
\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
\newcommand*{\@glo@no@assign@sortkey}[1]{%
  \PackageError{glossaries}{‘sort’ key not permitted with
  \string\printglossary}%
  {The ‘sort’ key may only be used with \string\printnoidxglossary}%
}
\newcommand*{\@glo@assign@sortkey}[1]{%
  \def\@glo@sorttype{#1}%
}
\newcommand*{\@glsnonextpages}{%
  \gdef\glossaryentrynumbers##1{%
    \glsresetentrylist
  }%

```

```

}
\newcommand*{\@glsnextpages}{%
  \gdef\glossaryentrynumbers##1{%
    ##1\glsresetentrylist}}
\newcommand*{\glsresetentrylist}{%
  \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
\newcommand*{\glsnonextpages}{}
\newcommand*{\glsnextpages}{}
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
\newcommand*{\glsresetsubentrycounter}{%
  \ifglssubentrycounter
    \setcounter{glossarysubentry}{0}%
  \fi
}
\newcommand*{\glsresetentrycounter}{%
  \ifglentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*{\glsstepentry}[1]{%
  \ifglentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
\newcommand*{\glsstepsubentry}[1]{%
  \ifglssubentrycounter
    \edef\currentglssubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
  \fi
}
\newcommand*{\glsrefentry}[1]{%
  \ifglentrycounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else
    \ifglssubentrycounter
      \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
      \gls{#1}%
    \fi
  \fi
}
\newcommand*{\glsentrycounterlabel}{%
  \ifglentrycounter
    \theglossaryentry.\space
  \fi
}
\newcommand*{\glssubentrycounterlabel}{%

```

```

\ifglssubentrycounter
  \theglossarysubentry)\space
\fi
}
\newcommand*\glssentryitem}[1]{%
  \ifglssentrycounter
    \glssstepentry{#1}\glssentrycounterlabel
  \else
    \glssresetsubentrycounter
  \fi
}
\newcommand*\glsssubentryitem}[1]{%
  \ifglsssubentrycounter
    \glssstepsubentry{#1}\glsssubentrycounterlabel
  \fi
}
\ifcsundef{theglossary}%
  {%
    \newenvironment{theglossary}{}{}%
  }%
  {%
    \@gls@warnontheglossdefined
    \renewenvironment{theglossary}{}{}%
  }
\newcommand*\glossaryheader{}
\newcommand*\glstarget}[2]{\@glstarget{\glolinkprefix#1}{#2}}
\providecommand*\compatibleglossentry}[2]{%
  \toks@{#2}%
  \protected@edef\@do@glossentry{\noexpand\glossaryentryfield{#1}%
    {\noexpand\glsnamefont
      {\expandafter\expandonce\csname glo@#1@name\endcsname}}%
    {\expandafter\expandonce\csname glo@#1@desc\endcsname}%
    {\expandafter\expandonce\csname glo@#1@symbol\endcsname}%
    {\the\toks@}}%
  }%
  \@do@glossentry
}
\newcommand*\glossentryname}[1]{%
  \glssdoifexistsorwarn{#1}%
  {%
    \letcs{\glo@name}{glo@\glssdetoklabel{#1}@name}%
    \expandafter\glsnamefont\expandafter{\glo@name}%
  }%
}
\newcommand*\Glossentryname}[1]{%
  \glssdoifexistsorwarn{#1}%
  {%
    \glsnamefont{\Glsentryname{#1}}%
  }%
}
}

```

```

\newcommand*\glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrydesc{#1}%
  }%
}
\newcommand*\Glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrydesc{#1}%
  }%
}
\newcommand*\glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrysymbol{#1}%
  }%
}
\newcommand*\Glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrysymbol{#1}%
  }%
}
\providecommand*\compatiblesubglossentry}[3]{%
  \toks@{#3}%
  \protected@edef\@do@subglossentry{\noexpand\glossarysubentryfield{\number#1}%
  {#2}%
  {\noexpand\glsnamefont
    {\expandafter\expandonce\csname glo@#2@name\endcsname}}%
  {\expandafter\expandonce\csname glo@#2@desc\endcsname}%
  {\expandafter\expandonce\csname glo@#2@symbol\endcsname}%
  {\the\toks@}}%
  }%
  \@do@subglossentry
}
\newcommand*\setglossentrycompatibility){%
  \let\glossentry\compatibleglossentry
  \let\subglossentry\compatiblesubglossentry
}
\setglossentrycompatibility
\newcommand*\glossaryentryfield}[5]{%
  \GlossariesWarning
  {Deprecated use of \string\glossaryentryfield.^~J
  I recommend you change to \string\glossentry.^~J
  If you've just upgraded, try removing your gls auxiliary
  files^~J and recompile}%
  \noindent\textbf{\glstarget{#1}{#2}} #4 #3. #5\par}
\newcommand*\glossarysubentryfield}[6]{%
  \GlossariesWarning

```

```

{Deprecated use of \string\glossarysubentryfield.^^J
 I recommend you change to \string\subglossentry.^^J
 If you've just upgraded, try removing your gls auxiliary
 files^^J and recompile}%
\glstarget{#2}{\strut}#4. #6\par}
\newcommand*{\glsgroupskip}{}
\newcommand*{\glsgroupheading}[1]{}
\newcommand*{\glsgetgrouptitle}[1]{%
 \@gls@getgrouptitle{#1}{\@gls@grptitle}%
 \@gls@grptitle
}
\newcommand*{\@gls@getgrouptitle}[2]{%
 \dtl@ifsingle{#1}%
 {%
 \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
 }%
 {%
 \ifboolexpr{test{\ifstrequal{#1}{\glsymbols}}
 or test{\ifstrequal{#1}{\glsnumbers}}}%
 {%
 \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
 }%
 {%
 \def#2{#1}%
 }%
 }%
}
}
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%
 \DTLifint{#1}%
 {\edef#2{\char#1\relax}}%
 {%
 \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
 }%
}
}
\newcommand*{\glsgetgrouplabel}[1]{%
 \ifthenelse{\equal{#1}{\glsymbolsgroupname}}{\glsymbols}{%
 \ifthenelse{\equal{#1}{\glsnumbersgroupname}}{\glsnumbers}{#1}}
\newcommand*{\setentrycounter}[2][]{%
 \def\@glo@counterprefix{#1}%
 \ifx\@glo@counterprefix\@empty
 \def\@glo@counterprefix{.}%
 \else
 \def\@glo@counterprefix{.#1}%
 \fi
 \def\glsentrycounter{#2}%
}
}
\newcommand*{\setglossarystyle}[1]{%
 \ifcsundef{@glsstyle@#1}%
 {%
 \PackageError{glossaries}{Glossary style '#1' undefined}{}%

```

```

}%
{%
  \csname @glsstyle@#1\endcsname
}%
\ifx\@glossary@default@style\relax
  \protected@edef\@glossary@default@style{#1}%
\fi
}
\newcommand*\@glossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \GlossariesWarning
    {Deprecated command \string\glossarystyle.^^J
     I recommend you switch to \string\setglossarystyle\space unless
     you want to maintain backward compatibility}%
    \setglossentrycompatibility
    \csname @glsstyle@#1\endcsname
    \ifcsdef{@glscompstyle@#1}%
    {\setglossentrycompatibility\cuse{@glscompstyle@#1}}%
    {}%
  }%
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\newcommand*\@newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
\newcommand*\@renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
\newcommand*\@glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\glsnumber#1{#1}%

```

```

}%
{%
  \def\glsnumber#1{\@glsnumber#1\nohyphenpage{}}\@nil}
}
\def\@glsnumber#1\@nohyphenpage#2#3\@nil{%
  \ifx\#1\%
  \else
    \@delimR#1\delimR\delimR\%
  \fi
  \ifx\#2\%
  \else
    #2%
  \fi
  \ifx\#3\%
  \else
    \@glsnumber#3\@nil
  \fi
}
\def\@delimR#1\delimR #2\delimR #3\%
\ifx\#2\%
  \@delimN{#1}%
\else
  \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
\fi}
\def\@delimN#1{\@delimN#1\delimN \delimN\}
\def\@delimN#1\delimN #2\delimN#3\%
\ifx\#3\%
  \@gls@numberlink{#1}%
\else
  \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
\fi
}
\def\@gls@numberlink#1{%
\begingroup
\toks@={}%
\@gls@removespaces#1 \@nil
\endgroup}
\def\@gls@removespaces#1 #2\@nil{%
\toks@=\expandafter{\the\toks@#1}%
\ifx\#2\%
  \edef\x{\the\toks@}%
  \ifx\x\empty
  \else
    \hyperlink{\glsentrycounter\@gls@counterprefix\the\toks@}%
      {\the\toks@}%
  \fi
\else
  \@gls@ReturnAfterFi{%
    \@gls@removespaces#2\@nil
  }%
}

```

```

\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
\newcommand*\hyperrm[1]{\textrm{\glsnumber{#1}}}
\newcommand*\hypersf[1]{\textsf{\glsnumber{#1}}}
\newcommand*\hypertt[1]{\texttt{\glsnumber{#1}}}
\newcommand*\hyperbf[1]{\textbf{\glsnumber{#1}}}
\newcommand*\hypermd[1]{\textmd{\glsnumber{#1}}}
\newcommand*\hyperit[1]{\textit{\glsnumber{#1}}}
\newcommand*\hypersl[1]{\textsl{\glsnumber{#1}}}
\newcommand*\hyperup[1]{\textup{\glsnumber{#1}}}
\newcommand*\hypersc[1]{\textsc{\glsnumber{#1}}}
\newcommand*\hyperemph[1]{\emph{\glsnumber{#1}}}
\newcommand{\oldacronym}[4][\gls@label]{%
  \def\gls@label{#2}%
  \newacronym[4]{#1}{#2}{#3}%
  \ifcsundef{xspace}%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}%
    }%
  }%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
        \noexpand\gls{#1}\noexpand\xspace}%
      }%
  }%
}
\newcommand{\newacronym}[4][{}]{
\newcommand*\acrpluralsuffix{\glsacrpluralsuffix}
\newrobustcmd*\glsstextup[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
\newcommand*\glsshortkey{short}
\newcommand*\glsshortpluralkey{shortplural}
\newcommand*\glslongkey{long}
\newcommand*\glslongpluralkey{longplural}
\newrobustcmd*\acrfull{\@gls@hyp@opt\ns@acrfull}
\newcommand*\ns@acrfull[2][{}]{%
  \new@ifnextchar[{\@acrfull{#1}{#2}}%
  {\@acrfull{#1}{#2} []}%
}
\def\@acrfull#1#2[#3]{%
  \acrfullfmt{#1}{#2}{#3}%
}
\newcommand*\acrfullfmt[3]{%
  \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newcommand{\acrlinkfullformat}[5]{%
  \acrfullformat{#1}{#3}{#4}{#5}{#2}{#3}{#4} []}%
}

```



```

\newcommand{\acrfullformat}[2]{#1\glsspace(#2)}
\newrobustcmd{\glsspace}{\space}
\newrobustcmd*{\Acrfull}{\@gls@hyp@opt\ns@Acrfull}
\newcommand*\ns@Acrfull[2][]{%
  \new@ifnextchar[{\@Acrfull{#1}{#2}}{%
    {\@Acrfull{#1}{#2} []}%
  }
}
\def\@Acrfull#1#2[#3]{%
  \Acrfullfmt{#1}{#2}{#3}%
}
\newcommand*{\Acrfullfmt}[3]{%
  \acrlinkfullformat{\@Acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfull}{\@gls@hyp@opt\ns@ACRfull}
\newcommand*\ns@ACRfull[2][]{%
  \new@ifnextchar[{\@ACRfull{#1}{#2}}{%
    {\@ACRfull{#1}{#2} []}%
  }
}
\def\@ACRfull#1#2[#3]{%
  \ACRfullfmt{#1}{#2}{#3}%
}
\newcommand*{\ACRfullfmt}[3]{%
  \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfullpl}{\@gls@hyp@opt\ns@acrfullpl}
\newcommand*\ns@acrfullpl[2][]{%
  \new@ifnextchar[{\@acrfullpl{#1}{#2}}{%
    {\@acrfullpl{#1}{#2} []}%
  }
}
\def\@acrfullpl#1#2[#3]{%
  \acrfullplfmt{#1}{#2}{#3}%
}
\newcommand*{\acrfullplfmt}[3]{%
  \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*{\Acrfullpl}{\@gls@hyp@opt\ns@Acrfullpl}
\newcommand*\ns@Acrfullpl[2][]{%
  \new@ifnextchar[{\@Acrfullpl{#1}{#2}}{%
    {\@Acrfullpl{#1}{#2} []}%
  }
}
\def\@Acrfullpl#1#2[#3]{%
  \Acrfullplfmt{#1}{#2}{#3}%
}
\newcommand*{\Acrfullplfmt}[3]{%
  \acrlinkfullformat{\@Acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfullpl}{\@gls@hyp@opt\ns@ACRfullpl}
\newcommand*\ns@ACRfullpl[2][]{%
  \new@ifnextchar[{\@ACRfullpl{#1}{#2}}{%
    {\@ACRfullpl{#1}{#2} []}%
  }
}

```

```

}
\def\ACRfullpl#1#2[#3]{%
  \ACRfullplfmt{#1}{#2}{#3}%
}
\newcommand*\ACRfullplfmt}[3]{%
  \acrlinkfullformat{\ACRlongpl}{\ACRshortpl}{#1}{#2}{#3}%
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand*\acrnameformat}[2]{\acronymfont{#1}}
\newtoks\glskeylisttok
\newtoks\glslabeltok
\newtoks\glsshorttok
\newtoks\gslongtok
\newcommand*\newacronymhook{}
\newcommand*\SetGenericNewAcronym}{%
  \let\@Gls@entryname\@Gls@acrenryname
  \renewcommand{\newacronym}[4][]{%
    \ifdefempty{\@glsacronymlists}%
      {%
        \def\@glo@type{\acronymtype}%
        \setkeys{glossentry}{##1}%
        \DeclareAcronymList{\@glo@type}%
      }%
    }%
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \gslongtok{##4}%
  \newacronymhook
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\expandonce{\acronymentry{##2}}},%
      sort={\acronymsort{\the\glsshorttok}{\the\gslongtok}},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\gslongtok},%
      longplural={\the\gslongtok\noexpand\acrpluralsuffix},%
      \GenericAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}%
\renewcommand*\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}%
\renewcommand*\Acrfullfmt}[3]{%

```

```

\glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}}%
\renewcommand*{\acrfullplfmt}[3]{%
\glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}}%
\renewcommand*{\glsentryfull}[1]{\genacrfullformat{##1}{}}}%
\renewcommand*{\Glsentryfull}[1]{\Genacrfullformat{##1}{}}}%
\renewcommand*{\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}}%
\renewcommand*{\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}}%
}
\newcommand*{\GenericAcronymFields}{description={\the\glslongtok}}
\newcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
\newcommand*{\acronymsort}[2]{##1}
\newcommand*{\setacronymstyle}[1]{%
\ifcsundef{@glsacr@dispstyle@##1}
{%
\PackageError{glossaries}{Undefined acronym style ‘##1’}{%
}%
}%
\ifdefempty{\@glsacronymlists}%
{%
\DeclareAcronymList{\acronymtype}%
}%
{}%
\SetGenericNewAcronym
\GlsUseAcrStyleDefs{##1}%
\@for\@gls@type:=\@glsacronymlists\do{%
\defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{##1}}%
}%
}%
}
\newcommand*{\newacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@##1}%
{%
\PackageError{glossaries}{Acronym style ‘##1’ already exists}{%
}%
}%
\csdef{@glsacr@dispstyle@##1}{##2}%
\csdef{@glsacr@styledefs@##1}{##3}%
}%
}
\newcommand*{\renewacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@##1}%
{%

```

```

\csdef{@glsacr@dispstyle@#1}{#2}%
\csdef{@glsacr@styledefs@#1}{#3}%
}%
{%
\PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
}%
}
\newcommand*{\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*{\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
{%
\ifglshaslong{\glslabel}{\glsacrfmt}{\glsacrfmt}}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
\ifglshaslong{\glslabel}{\glsacrfmt}{\glsacrfmt}}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}

```

```

}%
\renewcommand*\genplacrfullformat}[2]{%
  \glentrylongpl{##1}##2\glsacspace{##1}%
  (\protect\firstacronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
  \Glentrylongpl{##1}##2\glsacspace{##1}%
  (\protect\firstacronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newcommand*\glsacspace}[1]{%
  \settowidth{\dimen@}{(\protect\firstacronymfont{\glentryshort{##1}})}%
  \ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
  \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat}[2]{%
  \protect\firstacronymfont{\glentryshort{##1}}##2\space
  (\glentrylong{##1})%
}%
\renewcommand*\Genacrfullformat}[2]{%
  \protect\firstacronymfont{\Glentryshort{##1}}##2\space
  (\glentrylong{##1})%
}%
\renewcommand*\genplacrfullformat}[2]{%
  \protect\firstacronymfont{\glentryshortpl{##1}}##2\space
  (\glentrylongpl{##1})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glentryshortpl{##1}}##2\space
  (\glentrylongpl{##1})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{\%
  \GlsUseAcrEntryDispStyle{long-short}%
}

```

```

}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{long-sm-short}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{sc-short-long}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{sm-short-long}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
  \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{long-sp-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sp-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sp-short}%
}

```

```

\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glentrylong{##1}\glsacspace{##1}(\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sc-short-desc}%
{
  \GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{
  \GlsUseAcrStyleDefs{long-sc-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{long-sm-short-desc}%
{
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{short-long-desc}%
{
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sc-short-long-desc}%
{
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sm-short-long-desc}%

```

```

{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
  }
\newacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \Glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \mfirstucMakeUppercase
            {\glsentrylongpl{\glslabel}\glsinsert}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \Glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \mfirstucMakeUppercase
          {\glsentrylong{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glsgenentryfmt
  }%
  {\glscustomtext\glsinsert}%
}%

```



```

{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\glsentrylong{##2}##3\space}
  (\acronymfont{\glsentryshort{##2}})}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Glsentrylong{##2}##3\space}
  (\acronymfont{\glsentryshort{##2}})}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\glsentrylong{##2}##3\space}
    (\acronymfont{\glsentryshort{##2}})}%
  }%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\glsentrylongpl{##2}##3\space}
  (\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Glsentrylongpl{##2}##3\space}
  (\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\glsentrylongpl{##2}##3\space}
    (\acronymfont{\glsentryshortpl{##2}})}%
  }%
\renewcommand*{\glsentryfull}[1]{%
  \glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
  \glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
  \GlsUseAcrEntryDispStyle{dua}%
}%
{%
  \GlsUseAcrStyleDefs{dua}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentrylong{##1}}}%
  \renewcommand*{\acronymsort}[2]{##2}%
}%
\newacronymstyle{footnote}%

```

```

{%
\ifglshaslong{\glslabel}{\glsгенacfmt}{\glsгенentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshort{##1}}##2%
\protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\firstacronymfont{\Glsentryshort{##1}}##2%
\protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshortpl{##1}}##2%
\protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
\protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
(\glsentrylong{##2})}%
\renewcommand*{\Acrfullfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
(\glsentrylong{##2})}%
\renewcommand*{\ACRfullfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\acronymfont{\glsentryshort{##2}}##3\space
(\glsentrylong{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
(\glsentrylongpl{##2})}%
\renewcommand*{\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
(\glsentrylongpl{##2})}%
\renewcommand*{\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{##2}}##3\space
(\glsentrylongpl{##2})}}}%
\renewcommand*{\glsentryfull}[1]{%
\acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
\acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%

```

```

\renewcommand*\glentryfullpl}[1]{%
  \acronymfont{\glentryshortpl{##1}}\space(\glentrylongpl{##1})}%
\renewcommand*\Glentryfullpl}[1]{%
  \acronymfont{\Glentryshortpl{##1}}\space(\glentrylongpl{##1})}%
}
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
}%
\newacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*\acrpluralsuffix{\glsacrpluralsuffix}%
}%
\newacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*\{GenericAcronymFields}\{}%
  \renewcommand*\{acronymsort}[2]{##2}%
  \renewcommand*\{acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*\{GenericAcronymFields}\{}%
  \renewcommand*\{acronymsort}[2]{##2}%
  \renewcommand*\{acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}

```

```

}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newcommand*\DefineAcronymSynonyms{%
  \let\acs\acrshort
  \let\Acs\Acrshort
  \let\acsp\acrshortpl
  \let\Acsp\Acrshortpl
  \let\acl\aclong
  \let\aclp\aclongpl
  \let\Acl\Aclong
  \let\Aclp\Aclongpl
  \let\acf\acrfull
  \let\acfp\acrfullpl
  \let\Acf\Acrfull
  \let\Acfp\Acrfullpl
  \let\ac\gls
  \let\Ac\Gls
  \let\acp\glspl
  \let\Acp\Glspl
}
\ifglsacrshortcuts
  \DefineAcronymSynonyms
\fi
\newcommand*\SetDefaultAcronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsentryfmt}%
}
\newcommand*\DefaultNewAcronymDef{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      sort={\the\glsshorttok},%
      text={\the\glsshorttok},%
      first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
      plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      firstplural={\acrfullformat{\noexpand\expandonce\noexpand\@glo@longpl}%
        {\noexpand\expandonce\noexpand\@glo@shortpl}},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
    }
  }
}

```

```

        \the\glskeylisttok
    }%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@@gls@expand@field{##1}{descplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*\@SetDefaultAcronymStyle{%
  \@for\@gls@type:=\@gls@acronymlists\do{%
    \@SetDefaultAcronymDisplayStyle{\@gls@type}%
  }%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@gls@acronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \@SetDefaultAcronymDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \DefaultNewAcronymDef
  }%
  \renewcommand*\@acrpluralsuffix{\@gls@acrpluralsuffix}%
}
\newcommand*\@acrfootnote}[3]{\@acrlinkfootnote{#1}{#2}{#3}}
\newcommand*\@acrlinkfootnote}[3]{%
  \footnote{\@glslink[#1]{#2}{#3}}%
}
\newcommand*\@acrnofootnote}[3]{%
  \footnote{#3}%
}
\newcommand*\@SetDescriptionFootnoteAcronymDisplayStyle}[1]{%
  \def\glsentryfmt[#1]{%
    \ifdefempty\gls@customtext

```

```

{%
  \ifglused{\glslabel}%
  {%
    \acronymfont{\glsgenentryfmt}%
  }%
  {%
    \firstacronymfont{\glsgenentryfmt}%
    \ifglshassymbol{\glslabel}%
    {%
      \expandafter\protect\expandafter\acrfootnote\expandafter
      {\@gls@link@opts}{\@gls@link@label}%
      {%
        \glsifplural
        {\glsentrysymbolplural{\glslabel}}%
        {\glsentrysymbol{\glslabel}}%
      }%
    }%
  }%
  }%
  }%
  {\glscustomtext\glsinsert}%
}%
}
\newcommand*{\DescriptionFootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
  type=\acronymtype,%
  name={\noexpand\acronymfont{\the\glsshorttok}},%
  sort={\the\glsshorttok},%
  first={\the\glsshorttok},%
  firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
  text={\the\glsshorttok},%
  plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
  short={\the\glsshorttok},%
  shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
  long={\the\glslongtok},%
  longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
  symbol={\the\glslongtok},%
  symbolplural={\noexpand\expandonce\noexpand\@glo@longpl},%
  \the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
\@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
\@gls@expand@field{##1}{plural}{##2}%
}

```

```

}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*\SetDescriptionFootnoteAcronymStyle{%
  \renewcommand{\newacronym}[4][{}]{%
    \ifx\@glsacronymlists\@empty
      \def\@glo@type{acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetDescriptionFootnoteAcronymDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \DescriptionFootnoteNewAcronymDef
  }%
  \@for\@gls@type:=\@glsacronymlists\do{%
    \SetDescriptionFootnoteAcronymDisplayStyle{\@gls@type}%
  }%
  \ifglsacrsmallcaps
    \renewcommand*\acronymfont[1]{\textsc{##1}}%
    \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
  \else
    \ifglsacrsmaller
      \renewcommand*\acronymfont[1]{\textsmaller{##1}}%
    \fi
  \fi
  \ifglsacrdua
    \PackageError{glossaries}{Option clash: 'footnote' and 'dua'
      can't both be set}{}%
  \fi
}%
\newcommand*\SetDescriptionDUAAcronymDisplayStyle[1]{%
  \defglsentryfmt[##1]{\glsgenentryfmt}%
}
\newcommand*\DescriptionDUANewAcronymDef{%
  \edef\do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=acronymtype,%
      name={\the\glslongtok},%
      sort={\the\glslongtok},

```

```

text={\the\glslongtok},%
first={\the\glslongtok},%
plural={\noexpand\expandonce\noexpand\@glo@longpl},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
symbol={\the\glsshorttok},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionDUAAcronymStyle}{%
  \ifglsacrsmallcaps
    \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
      can’t both be set}{}%
  \else
    \ifglsacrsmaller
      \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
        can’t both be set}{}%
    \fi
  \fi
\renewcommand{\newacronym}[4][[]]{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetDescriptionDUAAcronymDisplayStyle{\@glo@type}%
  \fi
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%

```



```

\glslongtok{##4}%
\newacronymhook
\DescriptionDUANewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetDescriptionDUAACronymDisplayStyle{\@gls@type}%
}%
}%
\newcommand*\SetDescriptionAcronymDisplayStyle}[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\ifglsused{\glslabel}%
{%
\let\gls@org@insert\glsinsert
\let\glsinsert\@empty
\acronymfont{\glsentryfmt}\gls@org@insert
}%
}%
\glsentryfmt
\ifglsymbol{\glslabel}%
{%
\glsifplural
{%
\def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%
}%
{%
\def\@glo@symbol{\glsentrysymbol{\glslabel}}%
}%
\space(\protect\firstacronymfont
{\gls@symbol}
{\@glo@symbol}
{\mfirstucMakeUppercase{\@glo@symbol}})%
}%
}%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*\DescriptionNewAcronymDef){%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand
\acrnameformat{\the\glsshorttok}{\the\glslongtok}},%
sort={\the\glsshorttok},%
first={\the\glslongtok},%

```

```

firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
symbol={\noexpand\@glo@text},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionAcronymStyle}{%
\renewcommand{\newacronym}[4][]{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetDescriptionAcronymDisplayStyle{\@glo@type}%
  \fi
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \glslongtok{##4}%
  \newacronymhook
  \DescriptionNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
  \SetDescriptionAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
\renewcommand{\acronymfont}[1]{\textsc{##1}}
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
\else

```

```

\ifglsmaller
\renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
\fi
\fi
}%
\newcommand*{\SetFootnoteAcronymDisplayStyle}[1]{%
\defglentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\let\gls@org@insert\glsinsert
\let\glsinsert@empty
\ifglsused{\glslabel}%
{%
\acronymfont{\glsentryfmt}\gls@org@insert
}%
{%
\firstacronymfont{\glsentryfmt}\gls@org@insert
\ifglshaslong{\glslabel}%
{%
\expandafter\protect\expandafter\acrfootnote\expandafter
{\@gls@link@opts}{\@gls@link@label}%
}%
\glsifplural
{\glsentrylongpl{\glslabel}}%
{\glsentrylong{\glslabel}}%
}%
}%
}%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*{\FootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
first={\the\glsshorttok},%
firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\the\glslongtok},%
descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%

```

```

        \the\glskeylisttok
    }%
}%
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@@gls@expand@field{##1}{descplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@descplural\@org@gls@assign@descplural
}
\newcommand*{\SetFootnoteAcronymStyle}{%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@glsacronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetFootnoteAcronymDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \FootnoteNewAcronymDef
  }%
  \@for\@gls@type:=\@glsacronymlists\do{%
    \SetFootnoteAcronymDisplayStyle{\@gls@type}%
  }%
  \ifglsacrsmallcaps
    \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
    \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
  \else
    \ifglsacrsmaller
      \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
    \fi
  \fi
  \ifglsacrdua
    \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
    can’t both be set}{%
  \fi

```

```

}%
\DeclareRobustCommand*\glsdoparenifnotempty}[2]{%
  \protected@edef\gls@tmp{#1}%
  \ifdefempty\gls@tmp
  {}%
  {%
    \ifx\gls@tmp\@gls@default@value
    \else
    \space (#2{#1})%
    \fi
  }%
}
}
\newcommand*\SetSmallAcronymDisplayStyle}[1]{%
  \def\glsentryfmt[#1]{%
    \ifdefempty\gls@customtext
    {%
      \let\gls@org@insert\glsinsert
      \let\glsinsert\@empty
      \ifglsused{\glslabel}%
      {%
        \acronymfont{\gls@genentryfmt}\gls@org@insert
      }%
      {%
        \gls@genentryfmt
        \ifgls@hassymbol{\glslabel}%
        {%
          \gls@sifplural
          {%
            \def\@glo@symbol{\gls@entrysymbolplural{\glslabel}}%
          }%
          {%
            \def\@glo@symbol{\gls@entrysymbol{\glslabel}}%
          }%
          \space
          (\gls@scapscase
           {\firstacronymfont{\@glo@symbol}}%
           {\firstacronymfont{\@glo@symbol}}%
           {\firstacronymfont{\mfirstucMakeUppercase{\@glo@symbol}}})%
        }%
        {}%
      }%
    }%
    {\gls@customtext\glsinsert}%
  }%
}
\newcommand*\SmallNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{the\glslabeltok}%
    {%
      type=\acronymtype,%
    }
  }

```

```

name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
first={\the\glslongtok},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\noexpand\@glo@first},%
descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
symbol={\the\glsshorttok},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@@gls@expand@field{##1}{descplural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@descplural\@org@gls@assign@descplural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetSmallAcronymStyle}{%
\renewcommand{\newacronym}[4][{}]{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetSmallAcronymDisplayStyle{\@glo@type}%
  \fi
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
}

```

```

\glslongtok{##4}%
\newacronymhook
\SmallNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
  \SetSmallAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
  \renewcommand*\acronymfont[1]{\textsc{##1}}
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
\else
  \renewcommand*\acronymfont[1]{\textsmaller{##1}}
\fi
\ifglsacrdua
  \ifglsacrsmallcaps
    \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
      can’t both be set}{}%
  \else
    \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
      can’t both be set}{}%
  \fi
\fi
}%
\newcommand*\SetDUADisplayStyle[1]{%
  \defglsentryfmt[##1]{\glsentryfmt}%
}
\newcommand*\DUANewAcronymDef{%
  \edef\do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      text={\the\glslongtok},%
      first={\the\glslongtok},%
      plural={\noexpand\expandonce\noexpand\@glo@longpl},%
      firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      symbol={\the\glsshorttok},%
      symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      \the\glskeylisttok
    }%
  }%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural

```

```

\let\org@gl@assign@descplural\gl@assign@descplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@symbolplural##1##2{%
  \@gl@expand@field{##1}{symbolplural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@gl@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\org@gl@assign@firstpl
\let\gl@assign@plural\org@gl@assign@plural
\let\gl@assign@symbolplural\org@gl@assign@symbolplural
\let\gl@assign@descplural\org@gl@assign@descplural
}
\newcommand*{\SetDUASStyle}{%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@gl@sacronymlists\@empty
      \def\@gl@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@gl@type}%
      \SetDUADisplayStyle{\@gl@type}%
    \fi
    \gl@keylisttok{##1}%
    \gl@labeltok{##2}%
    \gl@shorttok{##3}%
    \gl@longtok{##4}%
    \newacronymhook
    \DUANewAcronymDef
  }%
  \@for\@gl@type:=\@gl@sacronymlists\do{%
    \SetDUADisplayStyle{\@gl@type}%
  }%
}
\newcommand*{\SetAcronymStyle}{%
  \SetDefaultAcronymStyle
  \ifgl@acr@description
    \ifgl@acr@footnote
      \SetDescriptionFootnoteAcronymStyle
    \else
      \ifgl@acr@dua
        \SetDescriptionDUAAcronymStyle
      \else
        \SetDescriptionAcronymStyle
      \fi
    \fi
  \fi
}

```



```

\else
  \ifglacrfootnote
    \SetFootnoteAcronymStyle
  \else
    \ifthenelse{\boolean{glacrsmallcaps}\OR
      \boolean{glacrsmaller}}{%
      {%
        \SetSmallAcronymStyle
      }%
    }%
    \ifglacrdua
      \SetDUASStyle
    \fi
  }%
\fi
}
\SetAcronymStyle
\newcommand*{\SetCustomDisplayStyle}[1]{%
  \defglentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*{\CustomAcronymFields}{%
  name={\the\glsshorttok},%
  description={\the\glslongtok},%
  first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
  firstplural={\acrfullformat
    {\noexpand\glsentrylongpl{\the\glslabeltok}}%
    {\noexpand\glsentryshortpl{\the\glslabeltok}}},%
  text={\the\glsshorttok},%
  plural={\the\glsshorttok\noexpand\acrpluralsuffix}%
}
\newcommand*{\CustomNewAcronymDef}{%
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=acronymtype,%
      short={\the\glsshorttok},%
      shortplural={\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},%
      \CustomAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}

```

```

\newcommand*\SetCustomStyle{%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@glsacronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetCustomDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \CustomNewAcronymDef
  }%
  \@for\@gls@type:=\@glsacronymlists\do{%
    \SetCustomDisplayStyle{\@gls@type}%
  }%
}
\RequirePackage{glossary-hypernav}[=v4.46]
\@gls@loadlist
\@gls@loadlong
\@gls@loadsuper
\@gls@loadtree
\ifx\@glossary@default@style\relax
\else
  \setglossarystyle{\@glossary@default@style}
\fi
\newcommand*\showgloparent}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
}
\newcommand*\showglolevel}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
}
\newcommand*\showglotext}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
}
\newcommand*\showgloplural}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@plural\endcsname
}
\newcommand*\showglofirst}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@first\endcsname
}
\newcommand*\showglofirstpl}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpl\endcsname
}
\newcommand*\showglotype}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@type\endcsname
}
\newcommand*\showglocounter}[1]{%

```

```

\expandafter\show\csname glo@\glsdetoklabel{#1}@counter\endcsname
}
\newcommand*{\showglouserii}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@userii\endcsname
}
\newcommand*{\showglouseriii}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@useriii\endcsname
}
\newcommand*{\showglouseriv}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@useriv\endcsname
}
\newcommand*{\showglouserv}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@userv\endcsname
}
\newcommand*{\showglouservi}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@uservi\endcsname
}
\newcommand*{\showgloname}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@name\endcsname
}
\newcommand*{\showglodesc}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@desc\endcsname
}
\newcommand*{\showglodescplural}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@descplural\endcsname
}
\newcommand*{\showglosort}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@sort\endcsname
}
\newcommand*{\showglosymbol}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@symbol\endcsname
}
\newcommand*{\showglosymbolplural}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@symbolplural\endcsname
}
\newcommand*{\showgloshort}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@short\endcsname
}
\newcommand*{\showglolong}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname
}
\newcommand*{\showgloindex}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname
}
\newcommand*{\showgloflag}[1]{%
\expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname
}
}

```

```

\newcommand*\showgloloclist}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@loclist\endcsname
}
\newcommand*\showglofield}[2]{%
  \csshow{glo@glstdetoklabel{#1}@#2}%
}
\newcommand*\showacronymlists}{%
  \show@glssacronymlists
}
\newcommand*\showglossaries}{%
  \show@glo@types
}
\newcommand*\showglossaryin}[1]{%
  \expandafter\show\csname @glotype@#1@in\endcsname
}
\newcommand*\showglossaryout}[1]{%
  \expandafter\show\csname @glotype@#1@out\endcsname
}
\newcommand*\showglossarytitle}[1]{%
  \expandafter\show\csname @glotype@#1@title\endcsname
}
\newcommand*\showglossarycounter}[1]{%
  \expandafter\show\csname @glotype@#1@counter\endcsname
}
\newcommand*\showglossaryentries}[1]{%
  \expandafter\show\csname glo@list@#1\endcsname
}
\csname ifglsc-compatible-2.07\endcsname
  \RequirePackage{glossaries-compatible-207}[v4.46]
\fi

```

8.2 Rollback v4.46 (glossaries-accsupp-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-accsupp}[2020/03/19 v4.46 (NLCT)
  Experimental glossaries accessibility]
\DeclareOption*\PassOptionsToPackage{\CurrentOption}{glossaries}}
\ProcessOptions
\@ifpackageloaded{glossaries-extra}
{%
  \ifx@glsxtr@doaccsupp\empty
  \GlossariesWarning{The ‘glossaries-accsupp’
    package has been loaded\MessageBreak
    after the ‘glossaries-extra’ package. This\MessageBreak
    can cause a failure to integrate both packages. \MessageBreak
    Either use the ‘accsupp’ option when you load\MessageBreak
    ‘glossaries-extra’ or load ‘glossaries-accsupp’\MessageBreak
    before loading ‘glossaries-extra’}%
}

```

```

\fi
}
{}
\def\compatibleglossentry#1#2{%
\ toks@{#2}%
\protected@edef\do@glossentry{%
\ noexpand\accsuppglossaryentryfield{#1}%
\ noexpand\glsnamefont
\ expandafter\expandonce\csname glo@\glsdetoklabel{#1}@name\endcsname}}%
\ expandafter\expandonce\csname glo@\glsdetoklabel{#1}@desc\endcsname}%
\ expandafter\expandonce\csname glo@\glsdetoklabel{#1}@symbol\endcsname}%
\ the\toks@}%
}%
\do@glossentry
}
\def\compatiblesubglossentry#1#2#3{%
\ toks@{#3}%
\protected@edef\do@subglossentry{%
\ noexpand\accsuppglossarysubentryfield{\number#1}%
\ #2}%
\ noexpand\glsnamefont
\ expandafter\expandonce\csname glo@\glsdetoklabel{#2}@name\endcsname}}%
\ expandafter\expandonce\csname glo@\glsdetoklabel{#2}@desc\endcsname}%
\ expandafter\expandonce\csname glo@\glsdetoklabel{#2}@symbol\endcsname}%
\ the\toks@}%
}%
\do@subglossentry
}
\RequirePackage{glossaries}[v4.46]
\providecommand{\gls@accsupp@engine}{accsupp}
\providecommand{\gls@accessibility}[4]{#4}
\ifdefstring\gls@accsupp@engine{accsupp}
{
\RequirePackage{accsupp}
\renewcommand{\gls@accessibility}[4]{%
\BeginAccSupp{#1,#2={#3}}#4\EndAccSupp{}}%
}
}
{}
\newcommand{\glsaccessibility}[4][[]]{%
\@glsshowaccsupp{#1}{#2}{#3}%
\gls@accessibility{#1}{#2}{#3}{#4}%
}
\define@key{glossentry}{access}{%
\def\@glo@access{#1}%
}
\define@key{glossentry}{textaccess}{%
\def\@glo@textaccess{#1}%
}
\define@key{glossentry}{firstaccess}{%

```

```

\def\@glo@firstaccess{#1}%
}
\define@key{glossentry}{pluralaccess}{%
\def\@glo@pluralaccess{#1}%
}
\define@key{glossentry}{firstpluralaccess}{%
\def\@glo@firstpluralaccess{#1}%
}
\define@key{glossentry}{symbolaccess}{%
\def\@glo@symbolaccess{#1}%
}
\define@key{glossentry}{symbolpluralaccess}{%
\def\@glo@symbolpluralaccess{#1}%
}
\define@key{glossentry}{descriptionaccess}{%
\def\@glo@descaccess{#1}%
}
\define@key{glossentry}{descriptionpluralaccess}{%
\def\@glo@descpluralaccess{#1}%
}
\define@key{glossentry}{shortaccess}{%
\def\@glo@shortaccess{#1}%
}
\define@key{glossentry}{shortpluralaccess}{%
\def\@glo@shortpluralaccess{#1}%
}
\define@key{glossentry}{longaccess}{%
\def\@glo@longaccess{#1}%
}
\define@key{glossentry}{longpluralaccess}{%
\def\@glo@longpluralaccess{#1}%
}
\define@key{glossentry}{user1access}{%
\def\@glo@useriaccess{#1}%
}
\define@key{glossentry}{user2access}{%
\def\@glo@useriiaccess{#1}%
}
\define@key{glossentry}{user3access}{%
\def\@glo@useriiiaccess{#1}%
}
\define@key{glossentry}{user4access}{%
\def\@glo@userivaccess{#1}%
}
\define@key{glossentry}{user5access}{%
\def\@glo@uservaccess{#1}%
}
\define@key{glossentry}{user6access}{%
\def\@glo@userviaccess{#1}%
}
}

```

```

\appto\@gls@keymap{,%
  {access}{access},%
  {textaccess}{textaccess},%
  {firstaccess}{firstaccess},%
  {pluralaccess}{pluralaccess},%
  {firstpluralaccess}{firstpluralaccess},%
  {symbolaccess}{symbolaccess},%
  {symbolpluralaccess}{symbolpluralaccess},%
  {descaccess}{descaccess},%
  {descpluralaccess}{descpluralaccess},%
  {shortaccess}{shortaccess},%
  {shortpluralaccess}{shortpluralaccess},%
  {longaccess}{longaccess},%
  {longpluralaccess}{longpluralaccess},%
  {user1access}{useriaccess},%
  {user2access}{useriiaccess},%
  {user3access}{useriiiaccess},%
  {user4access}{userivaccess},%
  {user5access}{uservaccess},%
  {user6access}{userviaccess}%
}
\def\@gls@noaccess{\relax}
\let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
\renewcommand*{\@newglossaryentryprehook}{%
  \@gls@oldnewglossaryentryprehook
  \def\@glo@access{\relax}%
  \def\@glo@textaccess{\@glo@access}%
  \def\@glo@firstaccess{\@glo@access}%
  \def\@glo@pluralaccess{\@glo@textaccess}%
  \def\@glo@firstpluralaccess{\@glo@pluralaccess}%
  \def\@glo@symbolaccess{\relax}%
  \def\@glo@symbolpluralaccess{\@glo@symbolaccess}%
  \def\@glo@descaccess{\relax}%
  \def\@glo@descpluralaccess{\@glo@descaccess}%
  \def\@glo@shortaccess{\relax}%
  \def\@glo@shortpluralaccess{\@glo@shortaccess}%
  \def\@glo@longaccess{\relax}%
  \def\@glo@longpluralaccess{\@glo@longaccess}%
  \def\@glo@useriaccess{\relax}%
  \def\@glo@useriiaccess{\relax}%
  \def\@glo@useriiiaccess{\relax}%
  \def\@glo@userivaccess{\relax}%
  \def\@glo@uservaccess{\relax}%
  \def\@glo@userviaccess{\relax}%
}
\let\@gls@oldnewglossaryentryposthook\@newglossaryentryposthook
\renewcommand*{\@newglossaryentryposthook}{%
  \@gls@oldnewglossaryentryposthook
  \expandafter
  \protected@xdef\csname glo@\@glo@label @access\endcsname{%

```

```

    \@glo@access}%
\expandafter
\protected@xdef\csname glo@\@glo@label @textaccess\endcsname{%
    \@glo@textaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstaccess\endcsname{%
    \@glo@firstaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @pluralaccess\endcsname{%
    \@glo@pluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstpluralaccess\endcsname{%
    \@glo@firstpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolaccess\endcsname{%
    \@glo@symbolaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolpluralaccess\endcsname{%
    \@glo@symbolpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descaccess\endcsname{%
    \@glo@descaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descpluralaccess\endcsname{%
    \@glo@descpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortaccess\endcsname{%
    \@glo@shortaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortpluralaccess\endcsname{%
    \@glo@shortpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longaccess\endcsname{%
    \@glo@longaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longpluralaccess\endcsname{%
    \@glo@longpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriaccess\endcsname{%
    \@glo@useriaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiaccess\endcsname{%
    \@glo@useriiaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiiaccess\endcsname{%
    \@glo@useriiiaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @userivaccess\endcsname{%
    \@glo@userivaccess}%
\expandafter

```



```

\protected@xdef\csname glo@\glo@label @servaccess\endcsname{%
  \@glo@servaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @serviaccess\endcsname{%
  \@glo@serviaccess}%
}
\newcommand*\glsentryaccess}[1]{%
  \@gls@entry@field{#1}{access}%
}
\newcommand*\glsentrytextaccess}[1]{%
  \@gls@entry@field{#1}{textaccess}%
}
\newcommand*\glsentryfirstaccess}[1]{%
  \@gls@entry@field{#1}{firstaccess}%
}
\newcommand*\glsentrypluralaccess}[1]{%
  \@gls@entry@field{#1}{pluralaccess}%
}
\newcommand*\glsentryfirstpluralaccess}[1]{%
  \@gls@entry@field{#1}{firstpluralaccess}%
}
\newcommand*\glsentrysymbolaccess}[1]{%
  \@gls@entry@field{#1}{symbolaccess}%
}
\newcommand*\glsentrysymbolpluralaccess}[1]{%
  \@gls@entry@field{#1}{symbolpluralaccess}%
}
\newcommand*\glsentrydescaccess}[1]{%
  \@gls@entry@field{#1}{descaccess}%
}
\newcommand*\glsentrydescpluralaccess}[1]{%
  \@gls@entry@field{#1}{descpluralaccess}%
}
\newcommand*\glsentryshortaccess}[1]{%
  \@gls@entry@field{#1}{shortaccess}%
}
\newcommand*\glsentryshortpluralaccess}[1]{%
  \@gls@entry@field{#1}{shortpluralaccess}%
}
\newcommand*\glsentrylongaccess}[1]{%
  \@gls@entry@field{#1}{longaccess}%
}
\newcommand*\glsentrylongpluralaccess}[1]{%
  \@gls@entry@field{#1}{longpluralaccess}%
}
\newcommand*\glsentryuseriaccess}[1]{%
  \@gls@entry@field{#1}{useriaccess}%
}
\newcommand*\glsentryuseriiaaccess}[1]{%
  \@gls@entry@field{#1}{useriiaaccess}%
}

```

```

}
\newcommand*\glseentryuseriiiaccess}[1]{%
  \@gls@entry@field{#1}{useriiiaccess}%
}
\newcommand*\glseentryuserivaccess}[1]{%
  \@gls@entry@field{#1}{userivaccess}%
}
\newcommand*\glseentryuservaccess}[1]{%
  \@gls@entry@field{#1}{uservaccess}%
}
\newcommand*\glseentryuserviaccess}[1]{%
  \@gls@entry@field{#1}{userviaccess}%
}
\newcommand{\glsfieldaccsupp}[4]{%
  \ifdef\glscategory
  {%
    \ifcsdef{glsxtr\glscategory{#4}#3accsupp}%
    {\csname glsxtr\glscategory{#4}#3accsupp\endcsname{#1}{#2}}%
    {%
      \ifcsdef{glsxtr\glscategory{#4}accsupp}%
      {\csname glsxtr\glscategory{#4}accsupp\endcsname{#1}{#2}}%
      {%
        \ifcsdef{gls#3accsupp}%
        {\csname gls#3accsupp\endcsname{#1}{#2}}%
        {\glsaccsupp{#1}{#2}}%
      }%
    }%
  }%
  {%
    \ifcsdef{gls#3accsupp}%
    {\csname gls#3accsupp\endcsname{#1}{#2}}%
    {\glsaccsupp{#1}{#2}}%
  }%
}
\newcommand{\xglsfieldaccsupp}[1]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsfieldaccsupp\expandafter{\@gls@replacementtext}%
}
\newcommand*\glsshortaccsupp}[2]{\glsaccessibility{E}{#1}{#2}}
\newcommand*\glsshortplaccsupp{\glsshortaccsupp}
\newcommand*\glsaccsupp}[2]{\glsaccessibility{ActualText}{#1}{#2}}
\newcommand*\xglsaccsupp}[2]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsaccsupp\expandafter{\@gls@replacementtext}{#2}%
}
\newcommand*\@gls@access@display}[2]{%
  \protected@edef\@glo@access{#2}%
  \ifx\@glo@access\@gls@noaccess
  #1%
  \else

```

```

        \xglsaccsupp{\@glo@access}{#1}%
    \fi
}
\newcommand*\@gls@fieldaccess@display[4]{%
    \protected@edef\@glo@access{#4}%
    \ifdefequal\@glo@access\@gls@noaccess
    {#3}%
    {\expandafter\glsfieldaccsupp\expandafter{\@glo@access}{#3}{#2}{#1}}%
}
\newrobustcmd*\@gls@nameaccessdisplay[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@access}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{name}{#1}{\glsentryaccess{#2}}}%
}
\newrobustcmd*\@gls@textaccessdisplay[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@textaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{text}{#1}{\glsentrytextaccess{#2}}}%
}
\newrobustcmd*\@gls@pluralaccessdisplay[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@pluralaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{plural}{#1}{\glsentrypluralaccess{#2}}}%
}
\newrobustcmd*\@gls@firstaccessdisplay[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@firstaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{first}{#1}{\glsentryfirstaccess{#2}}}%
}
\newrobustcmd*\@gls@firstpluralaccessdisplay[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@firstpluralaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{firstpl}{#1}{\glsentryfirstpluralaccess{#2}}}%
}
\newrobustcmd*\@gls@symbolaccessdisplay[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@symbolaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{symbol}{#1}{\glsentrysymbolaccess{#2}}}%
}
\newrobustcmd*\@gls@symbolpluralaccessdisplay[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@symbolpluralaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{symbolplural}{#1}{\glsentrysymbolpluralaccess{#2}}}%
}
\newrobustcmd*\@gls@descriptionaccessdisplay[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@descaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{desc}{#1}{\glsentrydescaccess{#2}}}%
}
\newrobustcmd*\@gls@descriptionpluralaccessdisplay[2]{%

```

```

\ifcsundef{glo@glstdetoklabel{#2}@descpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{descplural}{#1}{\glstentrydescpluralaccess{#2}}}%
}
\newrobustcmd*{\glsshortaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@shortaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{short}{#1}{\glstentryshortaccess{#2}}}%
}
\newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@shortpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glstentryshortpluralaccess{#2}}}%
}
\newrobustcmd*{\glslongaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@longaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{long}{#1}{\glstentrylongaccess{#2}}}%
}
\newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@longpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{longpl}{#1}{\glstentrylongpluralaccess{#2}}}%
}
\newrobustcmd*{\glsuseriaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@useriaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useri}{#1}{\glstentryuseriaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@useriiaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{userii}{#1}{\glstentryuseriiaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiiaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@useriiiaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useriii}{#1}{\glstentryuseriiiaccess{#2}}}%
}
\newrobustcmd*{\glsuserivaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@userivaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useriv}{#1}{\glstentryuserivaccess{#2}}}%
}
\newrobustcmd*{\glsuservaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@uservaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{userv}{#1}{\glstentryuservaccess{#2}}}%
}
\newrobustcmd*{\glsuserviaccessdisplay}[2]{%

```

```

\ifcsundef{glo@\glsdetoklabel{#2}@userviaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{uservi}{#1}{\glsentryuserviaccess{#2}}}%
}
\newrobustcmd*{\glsaccessdisplay}[3]{%
\ifcsundef{gls#1accessdisplay}%
{%
\PackageError{glossaries-accsupp}{No accessibility support
for key ‘#1’}{%
}%
}%
\csname gls#1accessdisplay\endcsname{#2}{#3}%
}%
}
\renewcommand*{\@gls@default@entryfmt}[2]{%
\ifdefempty\glscustomtext
{%
\glsifplural
{%
\glscapscase
{%
\ifglsused\glslabel
{%
#2{\glspluralaccessdisplay
{\glsentryplural{\glslabel}}{\glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glsentrydescplural{\glslabel}}{\glslabel}}%
{\glsymbolpluralaccessdisplay
{\glsentrysymbolplural{\glslabel}}{\glslabel}}
{\glsinsert}%
}%
}%
#1{\glsfirstpluralaccessdisplay
{\glsentryfirstplural{\glslabel}}{\glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glsentrydescplural{\glslabel}}{\glslabel}}%
{\glsymbolpluralaccessdisplay
{\glsentrysymbolplural{\glslabel}}{\glslabel}}%
{\glsinsert}%
}%
}%
}%
\ifglsused\glslabel
{%
#2{\glspluralaccessdisplay
{\Glsentryplural{\glslabel}}{\glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glsentrydescplural{\glslabel}}{\glslabel}}%
{\glsymbolpluralaccessdisplay
{\glsentrysymbolplural{\glslabel}}{\glslabel}}%
}
}
}

```

```

        {\glsinsert}%
    }%
    {%
        #1{\glsfirstpluralaccessdisplay
            {\Glsentryfirstplural{\glslabel}}{\glslabel}}%
            {\glsdescriptionpluralaccessdisplay
            {\glsentrydescplural{\glslabel}}{\glslabel}}%
            {\glsymbolpluralaccessdisplay
            {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
            {\glsinsert}%
        }%
    }%
    {%
        \ifglsused\glslabel
        {%
            \MakeUppercase{%
                #2{\glspluralaccessdisplay
                    {\glsentryplural{\glslabel}}{\glslabel}}%
                    {\glsdescriptionpluralaccessdisplay
                    {\glsentrydescplural{\glslabel}}{\glslabel}}%
                    {\glsymbolpluralaccessdisplay
                    {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
                    {\glsinsert}}%
            }%
        }%
        {%
            \MakeUppercase{%
                #1{\glsfirstpluralaccessdisplay
                    {\glsentryfirstplural{\glslabel}}{\glslabel}}%
                    {\glsdescriptionpluralaccessdisplay
                    {\glsentrydescplural{\glslabel}}{\glslabel}}%
                    {\glsymbolpluralaccessdisplay
                    {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
                    {\glsinsert}}%
            }%
        }%
    }%
    {%
        \glscapscase
        {%
            \ifglsused\glslabel
            {%
                #2{\glstextaccessdisplay
                    {\glsentrytext{\glslabel}}{\glslabel}}%
                    {\glsdescriptionaccessdisplay
                    {\glsentrydesc{\glslabel}}{\glslabel}}%
                    {\glsymbolaccessdisplay
                    {\glsentrysymbol{\glslabel}}{\glslabel}}%
                    {\glsinsert}%
                }%
            }%
        }%
    }%

```

```

#1{\glsfirstaccessdisplay
  {\glsentryfirst{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glsentrydesc{\glslabel}}{\glslabel}}%
  {\glsymbolaccessdisplay
    {\glsentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
}%
{%
\ifglsused\glslabel
{%
#2{\glstextaccessdisplay
  {\Glsentrytext{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glsentrydesc{\glslabel}}{\glslabel}}%
  {\glsymbolaccessdisplay
    {\glsentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
{%
#1{\glsfirstaccessdisplay
  {\Glsentryfirst{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glsentrydesc{\glslabel}}{\glslabel}}%
  {\glsymbolaccessdisplay
    {\glsentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
}%
{%
\ifglsused\glslabel
{%
\MakeUppercase{%
#2{\glstextaccessdisplay
  {\glsentrytext{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glsentrydesc{\glslabel}}{\glslabel}}%
  {\glsymbolaccessdisplay
    {\glsentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
{%
\MakeUppercase{%
#1{\glsfirstaccessdisplay
  {\glsentryfirst{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glsentrydesc{\glslabel}}{\glslabel}}%
  {\glsymbolaccessdisplay
    {\glsentrysymbol{\glslabel}}{\glslabel}}%

```



```

}%
{%
  \glsfirstpluralaccessdisplay
  {\Glsentryfirstplural{\glslabel}}{\glslabel}%
  \glsinsert
}%
}%
{%
  \ifglsused\glslabel
  {%
    \glspluralaccessdisplay
    {\mfirstucMakeUppercase{\glsentryplural{\glslabel}}}%
    {\glslabel}%
    \mfirstucMakeUppercase{\glsinsert}%
  }%
  {%
    \glsfirstpluralaccessdisplay
    {\mfirstucMakeUppercase{\glsentryfirstplural{\glslabel}}}%
    {\glslabel}%
    \mfirstucMakeUppercase{\glsinsert}%
  }%
  }%
}%
{%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      \glstextaccessdisplay{\glsentrytext{\glslabel}}{\glslabel}%
      \glsinsert
    }%
    {%
      \glsfirstaccessdisplay{\glsentryfirst{\glslabel}}{\glslabel}%
      \glsinsert
    }%
  }%
  {%
    \ifglsused\glslabel
    {%
      \glstextaccessdisplay{\Glsentrytext{\glslabel}}{\glslabel}%
      \glsinsert
    }%
    {%
      \glsfirstaccessdisplay{\Glsentryfirst{\glslabel}}{\glslabel}%
      \glsinsert
    }%
  }%
  }%
  {%
    \ifglsused\glslabel
    {%

```

```

        \glstextaccessdisplay
          {\mfirstucMakeUppercase{\glentrytext{\glslabel}}}{\glslabel}%
        \mfirstucMakeUppercase{\glsinsert}%
      }%
    {%
      \glsfirstaccessdisplay
        {\mfirstucMakeUppercase{\glentryfirst{\glslabel}}}{\glslabel}%
      \mfirstucMakeUppercase{\glsinsert}%
    }%
  }%
}
}
}
{%
  \glscustomtext\glsinsert
}
}
\renewcommand*{\glsgenacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglsused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \acronymfont
          {\glsshortpluralaccessdisplay
            {\glentryshortpl{\glslabel}}{\glslabel}}%
          \glsinsert
        }%
      }%
      \acronymfont
      {\glsshortpluralaccessdisplay
        {\Glsentryshortpl{\glslabel}}{\glslabel}}%
      \glsinsert
    }%
    {%
      \mfirstucMakeUppercase
      {\acronymfont
        {\glsshortpluralaccessdisplay
          {\glentryshortpl{\glslabel}}{\glslabel}}%
        \glsinsert}%
    }%
  }%
}
}
}
\glscapscase
{%
  \acronymfont
  {\glsshortaccessdisplay{\glentryshort{\glslabel}}{\glslabel}}%
  \glsinsert

```

```

    }%
    {%
      \acronymfont
      {\glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
      \glsinsert
    }%
    {%
      \mfirstucMakeUppercase
      {\acronymfont{%
        \glsshortaccessdisplay{\glsentryshort{\glslabel}}{\glslabel}}%
        \glsinsert}%
    }%
  }%
}
{%
\glsifplural
{%
  \glscapscase
  {%
    \genplacrfullformat{\glslabel}{\glsinsert}%
  }%
  {%
    \Genplacrfullformat{\glslabel}{\glsinsert}%
  }%
  {%
    \mfirstucMakeUppercase
    {\genplacrfullformat{\glslabel}{\glsinsert}}%
  }%
}
{%
  \glscapscase
  {%
    \genacrfullformat{\glslabel}{\glsinsert}%
  }%
  {%
    \Genacrfullformat{\glslabel}{\glsinsert}%
  }%
  {%
    \mfirstucMakeUppercase
    {\genacrfullformat{\glslabel}{\glsinsert}}%
  }%
}
}
}
\renewcommand*{\genacrfullformat}[2]{%
  \glslongaccessdisplay{\glsentrylong{#1}}{#1}#2\space

```

```

(\glsshortaccessdisplay{\protect\firstacronymfont{\glentryshort{#1}}}{#1})%
}
\renewcommand*{\Genacrfullformat}[2]{%
\glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space
(\glsshortaccessdisplay{\protect\firstacronymfont{\glentryshort{#1}}}{#1})%
}
\renewcommand*{\genplacrfullformat}[2]{%
\glslongpluralaccessdisplay{\glentrylongpl{#1}}{#1}#2\space
(\glshortpluralaccessdisplay
{\protect\firstacronymfont{\glentryshortpl{#1}}}{#1})%
}
\renewcommand*{\Genplacrfullformat}[2]{%
\glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}#2\space
(\glshortpluralaccessdisplay
{\protect\firstacronymfont{\glentryshortpl{#1}}}{#1})%
}
\def\@acrshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glsshortaccessdisplay{\glentryshort{#2}}{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@Acrshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@secondofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@ACRshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@thirdofthree

```

```

\let\glsinsert\@empty
\def\glscustomtext{%
  \acronymfont{\glsshortaccessdisplay
    {\MakeUppercase{\glsentryshort{#2}}}{#2}}#3%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{\glsentrylong{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@AcrLong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{\Glsentrylong{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@ACRLong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{%
        \MakeUppercase{\glsentrylong{#2}}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%

```

```

\glspostlinkhook
}
\def\@glstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\glstextentrytext{#2}}{#2}#3}%
}
\def\@GLstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\GLstextentrytext{#2}}{#2}#3}%
}
\def\@GLStext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glstextaccessdisplay{\mfirstucMakeUppercase{\glstextentrytext{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\glstextentryfirst{#2}}{#2}#3}%
}
\def\@Glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\GLstextentryfirst{#2}}{#2}#3}%
}
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsfirstaccessdisplay{\mfirstucMakeUppercase{\glstextentryfirst{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\glstextentryplural{#2}}{#2}#3}%
}
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\GLstextentryplural{#2}}{#2}#3}%
}
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glspluralaccessdisplay{\mfirstucMakeUppercase{\glstextentryplural{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\glstextentryfirstplural{#2}}{#2}#3}%
}
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\GLstextentryfirstplural{#2}}{#2}#3}%
}
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsfirstpluralaccessdisplay{\mfirstucMakeUppercase{\glstextentryfirstplural{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glstextentryname{#2}}{#2}#3}%
}
\def\@Glsname@#1#2[#3]{%

```

```

\@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\Glsentryname{#2}}{#2}#3}%
}
\def\@GLSname@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsnameaccessdisplay{\mfirstucMakeUppercase{\glsentryname{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@glsdesc@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\glsentrydesc{#2}}{#2}#3}%
}
\def\@Glsdesc@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
}
\def\@GLSdesc@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsdescriptionaccessdisplay{\mfirstucMakeUppercase{\glsentrydesc{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@glsdescplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\glsentrydescplural{#2}}{#2}#3}%
}
\def\@Glsdescplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
}
\def\@GLSdescplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsdescriptionpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrydescplural{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@glsymbol@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
}
\def\@Glsymbol@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}
\def\@GLSsymbol@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsymbolaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbol{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@glsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsymbolpluralaccessdisplay{\glsentrysymbolplural{#2}}{#2}#3}%
}
\def\@Glsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsymbolpluralaccessdisplay{\Glsentrysymbolplural{#2}}{#2}#3}%
}
\def\@GLSsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsymbolpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbolplural{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}

```

```

}
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glsentryuseri{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriaccessdisplay{\mfirstucMakeUppercase{\glsentryuseri{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glsentryuserii{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glsentryuserii{#2}}{#2}#3}%
}
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiaccessdisplay{\mfirstucMakeUppercase{\glsentryuserii{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\glsentryuseriii{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glsentryuseriii{#2}}{#2}#3}%
}
\def\@GLSuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiiaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriii{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glsentryuseriv{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
}
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuserivaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriv{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\glsentryuserv{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\Glsentryuserv{#2}}{#2}#3}%
}

```



```

}
\def\@GLSuser@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuseraccessdisplay{\mfirstucMakeUppercase{\glsentryuser{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glsentryuseri{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuseriaccessdisplay{\mfirstucMakeUppercase{\glsentryuseri{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
}
\renewcommand*\@glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\glsentryname{#1}}{#1}}%
  }%
}
\renewcommand*\@glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\Glsentryname{#1}}{#1}}%
  }%
}
\renewcommand*\@glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\glsentrydesc{#1}}{#1}%
  }%
}
\renewcommand*\@Glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}
\renewcommand*\@glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsymbolaccessdisplay{\glsentrysymbol{#1}}{#1}%
  }%
}
\renewcommand*\@Glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%

```

```

        \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
    }%
}
\newcommand*\accsuppglossaryentryfield}[5]{%
    \glossaryentryfield{#1}%
    {\glsnameaccessdisplay{#2}{#1}}%
    {\glsdescriptionaccessdisplay{#3}{#1}}%
    {\glssymbolaccessdisplay{#4}{#1}}{#5}%
}
\newcommand*\accsuppglossarysubentryfield}[6]{%
    \glossarysubentryfield{#1}{#2}%
    {\glsnameaccessdisplay{#3}{#2}}%
    {\glsdescriptionaccessdisplay{#4}{#2}}%
    {\glssymbolaccessdisplay{#5}{#2}}{#6}%
}
\renewacronymstyle{long-short}%
{%
    \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
    \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
    \renewcommand*\Genacrfullformat}[2]{%
        \glslongaccessdisplay{\glsentrylong{##1}}{##1}##2\space
        (\glsshortaccessdisplay
            {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
    }%
    \renewcommand*\Genacrfullformat}[2]{%
        \glslongaccessdisplay{\Glsentrylong{##1}}{##1}##2\space
        (\glsshortaccessdisplay
            {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
    }%
    \renewcommand*\Genplacrfullformat}[2]{%
        \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}##2\space
        (\glsshortpluralaccessdisplay
            {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
    }%
    \renewcommand*\Genplacrfullformat}[2]{%
        \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}##2\space
        (\glsshortpluralaccessdisplay
            {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
    }%
    \renewcommand*\acronymentry}[1]{%
        \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}
    }
    \renewcommand*\acronymsort}[2]{##1}%
    \renewcommand*\acronymfont}[1]{##1}%
    \renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
    \renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\renewacronymstyle{short-long}%
{%

```

```

\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2\space
(\glslongaccessdisplay{\glsentrylong{##1}}{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\Glsentryshort{##1}}}{##1}##2\space
(\glslongaccessdisplay{\glsentrylong{##1}}{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2\space
(\glslongpluralaccessdisplay
{\glsentrylongpl{##1}}{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2\space
(\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\renewacronymstyle{long-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}
\renewacronymstyle{long-sc-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sc-short}%
}

```

```

\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%

```

```

\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glslongpluralaccessdisplay{\glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay{\Glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay
            {\mfirstucMakeUppercase{\glsentrylongpl{\glslabel}}}{\glslabel}%
            \mfirstucMakeUppercase{\glsinsert}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \glslongaccessdisplay{\glsentrylong{\glslabel}}{\glslabel}\glsinsert
      }%
      {%
        \glslongaccessdisplay{\Glsentrylong{\glslabel}}{\glslabel}\glsinsert
      }%
      {%
        \glslongaccessdisplay
          {\mfirstucMakeUppercase
            {\glsentrylong{\glslabel}\glsinsert}}{\glslabel}%
          \mfirstucMakeUppercase{\glsinsert}%
      }%
    }%
  }%
  {%
    \glsgenentryfmt
  }%
  {\glscustomtext\glsinsert}%
}%
{%

```

```

\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay
    {\mfirstucMakeUppercase{\glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\Glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\mfirstucMakeUppercase{\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\glsentryfull}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
  \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}\space
  (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space
  (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}%
}

```

```

\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\renewacronymstyle{dua-desc}%
{%
\GlsUseAcrEntryDispStyle{dua}%
}%
{%
\GlsUseAcrStyleDefs{dua}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymentry}[1]{%
\glslongaccessdisplay{\acronymfont{\glsentrylong{##1}}}{##1}}%
\renewcommand*\acronymsort}[2]{##2}%
}%
\renewacronymstyle{footnote}%
{%
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*\genacrfullformat}[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}}{##1}}%
}%
\renewcommand*\Genacrfullformat}[2]{%
\glsshortaccessdisplay
{\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}}{##1}}%
}%
\renewcommand*\genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%
}%
\renewcommand*\Genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%
}%
\renewcommand*\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
\renewcommand*\acrfullfmt}[3]{%
\glslink[##1]{##2}{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}##3\space

```

```

        (\glslongaccessdisplay{\glsentrylong{##2}{##2}})}%
\renewcommand*\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##2}}}{##2}##3\space
        (\glslongaccessdisplay{\glsentrylong{##2}{##2}})}%
\renewcommand*\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortaccessdisplay
        {\mfirstucMakeUppercase
         {\acronymfont{\glsentryshort{##2}}}{##2}##3\space
         (\glslongaccessdisplay{\glsentrylong{##2}{##2}})}}%
\renewcommand*\acrfullplfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortpluralaccessdisplay
        {\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
        (\glslongpluralaccessdisplay{\glsentrylongpl{##2}{##2}})}%
\renewcommand*\Acrfullplfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortpluralaccessdisplay
        {\acronymfont{\Glsentryshortpl{##2}}}{##2}##3\space
        (\glslongpluralaccessdisplay{\glsentrylongpl{##2}})}%
\renewcommand*\ACRfullplfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortpluralaccessdisplay
        {\mfirstucMakeUppercase
         {\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
         (\glslongpluralaccessdisplay{\glsentrylongpl{##2}{##2}})}}%
\renewcommand*\glsentryfull}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}\space
    (\glslongaccessdisplay{\glsentrylong{##1}{##1}})%
\renewcommand*\Glsentryfull}[1]{%
    \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##1}}}{##1}\space
    (\glslongaccessdisplay{\glsentrylong{##1}{##1}})%
\renewcommand*\glsentryfullpl}[1]{%
    \glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##1}}}{##1}\space
    (\glslongpluralaccessdisplay{\glsentrylongpl{##1}{##1}})%
\renewcommand*\Glsentryfullpl}[1]{%
    \glsshortpluralaccessdisplay
    {\acronymfont{\Glsentryshortpl{##1}}}{##1}\space
    (\glslongpluralaccessdisplay{\glsentrylongpl{##1}{##1}})%
}
\renewacronymstyle{footnote-sc}%
{%
    \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
    \GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}

```



```

\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glstextup{\glspluralsuffix}}%
}%
\renewacronymstyle{footnote-sm}%
{%
\GlsUseAcrEntryDispStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}%
\renewacronymstyle{footnote-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sc-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
\GlsUseAcrStyleDefs{footnote-sc}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sm-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote-sm}%
}%
{%
\GlsUseAcrStyleDefs{footnote-sm}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
}

```

```

\newcommand*\glsdefaultshortaccess}[2]{#1}
\renewcommand*\newacronymhook{%
  \edef\@gls@keylist{%
    shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    shortpluralaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    \the\glskeylisttok}%
  \expandafter\glskeylisttok\expandafter{\@gls@keylist}%
}
\renewcommand*\DefaultNewAcronymDef{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      description={\the\glslongtok},%
      descriptionaccess=\relax,
      text={\the\glsshorttok},%
      access={\noexpand\@glo@textaccess},%
      sort={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      first={\noexpand\glslongaccessdisplay
        {\the\glslongtok}{\the\glslabeltok}\space
        (\noexpand\glsshortaccessdisplay
          {\the\glsshorttok}{\the\glslabeltok})},%
      plural={\the\glsshorttok\acrpluralsuffix},%
      firstplural={\noexpand\glslongpluralaccessdisplay
        {\noexpand\@glo@longpl}{\the\glslabeltok}\space
        (\noexpand\glsshortpluralaccessdisplay
          {\noexpand\@glo@shortpl}{\the\glslabeltok})},%
      firstaccess=\relax,
      firstpluralaccess=\relax,
      textaccess={\noexpand\@glo@shortaccess},%
      \the\glskeylisttok
    }%
  }%
  \let\@org@gls@assign@firstpl\gls@assign@firstpl
  \let\@org@gls@assign@plural\gls@assign@plural
  \let\@org@gls@assign@descplural\gls@assign@descplural
  \def\gls@assign@firstpl##1##2{%
    \@gls@expand@field{##1}{firstpl}{##2}%
  }%
  \def\gls@assign@plural##1##2{%
    \@gls@expand@field{##1}{plural}{##2}%
  }%
  \def\gls@assign@descplural##1##2{%

```

```

    \@gls@expand@field{##1}{descplural}{##2}%
  }%
  \@do@newglossaryentry
  \let\gls@assign@firstpl\@org@gls@assign@firstpl
  \let\gls@assign@plural\@org@gls@assign@plural
  \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\DescriptionFootnoteNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\noexpand\acronymfont{\the\glsshorttok}},%
      sort={\the\glsshorttok},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      access={\noexpand\@glo@textaccess},%
      plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      symbol={\the\glslongtok},%
      symbolplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      firstpluralaccess=\relax,
      textaccess={\noexpand\@glo@shortaccess},%
      \the\glskeylisttok
    }%
  }%
  \let\@org@gls@assign@firstpl\gls@assign@firstpl
  \let\@org@gls@assign@plural\gls@assign@plural
  \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
  \def\gls@assign@firstpl##1##2{%
    \@gls@expand@field{##1}{firstpl}{##2}%
  }%
  \def\gls@assign@plural##1##2{%
    \@gls@expand@field{##1}{plural}{##2}%
  }%
  \def\gls@assign@symbolplural##1##2{%
    \@gls@expand@field{##1}{symbolplural}{##2}%
  }%
  \@do@newglossaryentry
  \let\gls@assign@plural\@org@gls@assign@plural
  \let\gls@assign@firstpl\@org@gls@assign@firstpl
  \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\DescriptionNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%

```

```

type=\acronymtype,%
name={\noexpand
  \acronymformat{\the\glsshorttok}{\the\glslongtok}},%
access={\noexpand\@glo@textaccess},%
sort={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
first={\the\glslongtok},%
firstaccess=\relax,
firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
text={\the\glsshorttok},%
textaccess={\the\glslongtok},%
plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
symbol={\noexpand\@glo@text},%
symbolaccess={\noexpand\@glo@textaccess},%
symbolplural={\noexpand\@glo@plural},%
firstpluralaccess=\relax,
textaccess={\noexpand\@glo@shortaccess},%
\the\glskeylisttok}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\FootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\glsshorttok}},%
    sort={\the\glsshorttok},%
    text={\the\glsshorttok},%
    textaccess={\the\glslongtok},%
    access={\noexpand\@glo@textaccess},%

```

```

plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\the\glslongtok},%
descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
\the\glskeylisttok
}%
}%
\let\org@gls@assign@plural\gls@assign@plural
\let\org@gls@assign@firstpl\gls@assign@firstpl
\let\org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@gls@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\org@gls@assign@plural
\let\gls@assign@firstpl\org@gls@assign@firstpl
\let\gls@assign@descplural\org@gls@assign@descplural
}
\renewcommand*\SmallNewAcronymDef{%
\edef\do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\glsshorttok}},%
    access={\noexpand\@glo@symbolaccess},%
    sort={\the\glsshorttok},%
    short={\the\glsshorttok},%
    shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
    shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    long={\the\glslongtok},%
    longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    text={\noexpand\@glo@short},%
    textaccess={\noexpand\@glo@shortaccess},%
    plural={\noexpand\@glo@shortpl},%
    first={\the\glslongtok},%
    firstaccess=\relax,
    firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    description={\noexpand\@glo@first},%
    descriptionplural={\noexpand\@glo@firstplural},%
    symbol={\the\glsshorttok},%
    symbolaccess={\the\glslongtok},%

```

```

        symbolplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
        \the\glskeylisttok
    }%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
    \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
    \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
    \@@gls@expand@field{##1}{descplural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
    \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@descplural\@org@gls@assign@descplural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\glsshortaccesskey}{\glsshortkey access}%
\newcommand*{\glsshortpluralaccesskey}{\glsshortpluralkey access}%
\newcommand*{\glslongaccesskey}{\glslongkey access}%
\newcommand*{\glslongpluralaccesskey}{\glslongpluralkey access}%
\newcommand*{\showglongnameaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@access\endcsname
}
\newcommand*{\showglongtextaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@textaccess\endcsname
}
\newcommand*{\showglongpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@pluralaccess\endcsname
}
\newcommand*{\showglongfirstaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@firstaccess\endcsname
}
\newcommand*{\showglongfirstpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpluralaccess\endcsname
}
\newcommand*{\showglongsymbolaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolaccess\endcsname
}
\newcommand*{\showglongsymbolpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolpluralaccess\endcsname
}

```

```

}
\newcommand*{\showglodescaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@descaccess\endcsname
}
\newcommand*{\showglodescpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@descpluralaccess\endcsname
}
\newcommand*{\showgloshortaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@shortaccess\endcsname
}
\newcommand*{\showgloshortpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@shortpluralaccess\endcsname
}
\newcommand*{\showglolongaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@longaccess\endcsname
}
\newcommand*{\showglolongpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@longpluralaccess\endcsname
}
}

```

8.3 Rollback v4.46 (glossaries-babel-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-babel}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
  }%
}
}
}

```

8.4 Rollback v4.46 (glossaries-compatible-207-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-207}[2020/03/19 v4.46 (NLCT)]
\ifglxindy
  \renewcommand*\GlsAddXdyAttribute[1]{%
    \edef\xdyattributes{\xdyattributes ^^J \string"#1\string"}%
    \expandafter\toks@\expandafter{\xdylocref}%
    \edef\xdylocref{\the\toks@ ^^J%
    (markup-locref
    :open \string"\string~\n\string\setentrycounter
      {\noexpand\glscounter}%
      \expandafter\string\csname#1\endcsname
      \expandafter@gobble\string\{\string" ^^J
    :close \string"\expandafter@gobble\string\}\string" ^^J
    :attr \string"#1\string")}}
\fi
\renewcommand*\GlsAddXdyCounters[1]{%
  \GlossariesWarning{\string\GlsAddXdyCounters\space not available
    in compatibility mode.}%
}
\GlsAddXdyAttribute{glsnumberformat}
\GlsAddXdyAttribute{textrm}
\GlsAddXdyAttribute{textsf}
\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glsnumber}
\GlsAddXdyAttribute{hyperm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hypertt}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}
\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}
\ifglxindy
  \renewcommand*\GlsAddXdyLocation[2]{%
    \edef\xdyuserlocationdefs{%
      \xdyuserlocationdefs ^^J%
      (define-location-class \string"#1\string"^^J\space\space
        \space(#2))
  }

```



```

    }%
    \edef\xdyuserlocationnames{%
        \xdyuserlocationnames^^J\space\space\space
        \string"#1\string"}%
}
\fi
\renewcommand{\do@wrglossary}[1]{%
\ifglxsindy
    \expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
    \def\@glo@range{%
    \expandafter\if\@glo@prefix(\relax
        \def\@glo@range{:open-range}%
    \else
        \expandafter\if\@glo@prefix)\relax
        \def\@glo@range{:close-range}%
    \fi
    \fi
    \protected@edef\@glslocref{\theglentrycounter}%
    \@gls@checkmkidxchars\@glslocref
    \glossary[\csname glo@#1@type\endcsname]{%
    (indexentry :tkey (\csname glo@#1@index\endcsname)
        :locref \string"\@glslocref\string" %
        :attr \string"\@glo@suffix\string" \@glo@range
    )
    }%
\else
    \@set@glo@numformat\@glo@numfmt\@gls@counter\@glsnumberformat
    \glossary[\csname glo@#1@type\endcsname]{%
    \string\glossaryentry{\csname glo@#1@index\endcsname
        \@gls@encapchar\@glo@numfmt}{\theglentrycounter}}%
\fi
}
\def\@set@glo@numformat#1#2#3{%
    \expandafter\@glo@check@mkidxrangechar#3\@nil
    \protected@edef#1{%
        \@glo@prefix setentrycounter[]#{2}%
        \expandafter\string\csname\@glo@suffix\endcsname
    }%
    \@gls@checkmkidxchars#1%
}
\ifglxsindy
\def\writeist{%
    \openout\glswrite=\listfilename
    \write\glswrite{;; xindy style file created by the glossaries
        package in compatible-2.07 mode}%
    \write\glswrite{;; for document '\jobname' on
        \the\year-\the\month-\the\day}%
    \write\glswrite{^^J; required styles^^J}
    \@for\xdstyle:=\xdyrequiredstyles\do{%
        \if\xdstyle\@empty

```

```

\else
  \protected@write\glswrite{}{(require
    \string"\@xdystyle.xdy\string")}%
\fi
}%
\write\glswrite{^^J%
  ; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%
\protected@edef\@gls@roman{\@roman{0}\string"
  \string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@gls@roman
\edef\@tmp{\string" \string"roman-numbers-lowercase\string"
  :sep \string"}%
\@onelevel@sanitize\@tmp
\ifx\@tmp\@gls@roman
  \write\glswrite{(define-location-class
    \string"roman-page-numbers\string"^^J\space\space\space
    (\string"roman-numbers-lowercase\string")
    :min-range-length \@glsminrange)}%
\else
  \write\glswrite{(define-location-class
    \string"roman-page-numbers\string"^^J\space\space\space
    (:sep "\@gls@roman")
    :min-range-length \@glsminrange)}%
\fi
\write\glswrite{(define-location-class
  \string"Roman-page-numbers\string"^^J\space\space\space
  (\string"roman-numbers-uppercase\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"arabic-page-numbers\string"^^J\space\space\space
  (\string"arabic-numbers\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"alpha-page-numbers\string"^^J\space\space\space
  (\string"alpha\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"Alpha-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"Appendix-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string"
  :sep \string"\@glsAlphacompositor\string"
  \string"arabic-numbers\string")
  :min-range-length \@glsminrange)}%

```

```

\write\glswrite{(define-location-class
  \string"arabic-section-numbers\string"^^J\space\space\space
  (\string"arabic-numbers\string"
   :sep \string"\glscompositor\string"
   \string"arabic-numbers\string")
   :min-range-length \@glsminrange)}}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\glsseeformat\string"
  :close \string"{}\string")}%
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
  :open \string"\string
  \glossarysection[\string\glossarytoctitle]{\string
  \glossarytitle}\string\glossarypreamble\string~n\string\begin
  {theglossary}\string\glossaryheader\string~n\string" ^^J\space
  \space\space:close \string"\expandafter@gobble
  \string~%\string~n\string
  \end{theglossary}\string\glossarypostamble
  \string~n\string" ^^J\space\space\space
  :tree)}}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\string~n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \string~n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list
  :sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
  :sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixF" :length 1 :ignore-end)}%
\fi

```

```

\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
  :sep \string\string\glsgroupskip\string~n\string)}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
  :open-head \string\string\glsgroupheading
  \glsopenbrace\string^^J\space\space\space
  :close-head \string\glsclosebrace\string)}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}%
\write\glswrite{\@xdysortrules}%
\noist}
\else
\edef\gls@actualchar{\string?}
\edef\gls@encapchar{\string|}
\edef\gls@levelchar{\string!}
\edef\gls@quotechar{\string"}
\def\writeist{\relax
  \openout\glswrite=\istfilename
  \write\glswrite{\expandafter\@gobble\string\% makeindex style file
    created by the glossaries package}
  \write\glswrite{\expandafter\@gobble\string\% for document
    '\jobname' on \the\year-\the\month-\the\day}
  \write\glswrite{actual '\gls@actualchar'}
  \write\glswrite{encap '\gls@encapchar'}
  \write\glswrite{level '\gls@levelchar'}
  \write\glswrite{quote '\gls@quotechar'}
  \write\glswrite{keyword \string\string\glossaryentry\string"}
  \write\glswrite{preamble \string\string\glossarysection[\string
    \glossarytoctitle]{\string\glossarytitle}\string
    \glossarypreamble\string\n\string\begin{theglossary}\string
    \glossaryheader\string\n\string"}
  \write\glswrite{postamble \string\string\string\n\string
    \end{theglossary}\string\glossarypostamble\string\n
    \string"}
  \write\glswrite{group_skip \string\string\glsgroupskip\string\n
    \string"}
  \write\glswrite{item_0 \string\string\string\n\string"}
  \write\glswrite{item_1 \string\string\string\n\string"}
  \write\glswrite{item_2 \string\string\string\n\string"}
  \write\glswrite{item_01 \string\string\string\n\string"}
  \write\glswrite{item_x1

```

```

\string\string\relax \string\glsresetentrylist\string\n
\string}
\write\glswrite{item_12 \string\string\%\string\n\string}
\write\glswrite{item_x2
\string\string\relax \string\glsresetentrylist\string\n
\string}
\write\glswrite{delim_0 \string\string\{\string
\glossaryentrynumbers\string\{\string\relax \string}
\write\glswrite{delim_1 \string\string\{\string
\glossaryentrynumbers\string\{\string\relax \string}
\write\glswrite{delim_2 \string\string\{\string
\glossaryentrynumbers\string\{\string\relax \string}
\write\glswrite{delim_t \string\string\}\string}\string}
\write\glswrite{delim_n \string\string\delimN \string}
\write\glswrite{delim_r \string\string\delimR \string}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
\string\string\glsgroupheading\string\{\string}
\write\glswrite{heading_suffix
\string\string\}\string\relax
\string\glsresetentrylist \string}
\write\glswrite{symhead_positive \string"glssymbols\string}
\write\glswrite{numhead_positive \string"glnumbers\string}
\write\glswrite{page_compositor \string"glscpositor\string}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{suffix_2p \string"\gls@suffixF\string}
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{suffix_3p \string"\gls@suffixFF\string}
\fi
\noist
}
\fi
\renewcommand*{\noist}{\let\writeist\relax}

```

8.5 Rollback v4.46 (glossaries-compatible-307-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-307}[2020/03/19 v4.46 (NLCT)]
\newcommand{\compatglossarystyle}[2]{%
\ifcsundef{@glscompstyle@#1}%
{%
\csdef{@glscompstyle@#1}{#2}%
}%
}

```

```

    {%
      \PackageError{glossaries}{Glossary compatibility style ‘#1’ is already defined}{}%
    }%
  }
\compatglossarystyle{inline}{%
  \renewcommand{\glossaryentryfield}[5]{%
    \glsinlinedopostchild
    \gls@inlinesep
    \def\glo@desc{##3}%
    \def\@no@post@desc{\nopostdesc}%
    \glsentryitem{##1}\glsinlinenameformat{##1}{##2}%
    \ifx\glo@desc\@no@post@desc
      \glsinlineemptydescformat{##4}{##5}%
    \else
      \ifstrempy{##3}%
        {\glsinlineemptydescformat{##4}{##5}}%
        {\glsinlinedescformat{##3}{##4}{##5}}%
      \fi
      \ifglshaschildren{##1}%
        {%
          \glsresetsubentrycounter
          \glsinlineparentchildseparator
          \def\gls@inlinesubsep{}%
          \def\gls@inlinepostchild{\glsinlinepostchild}%
        }%
        {}%
      \def\gls@inlinesep{\glsinlineseparator}%
    }%
  \renewcommand{\glossarysubentryfield}[6]{%
    \gls@inlinesubsep%
    \glsinlinesubnameformat{##2}{##3}%
    \glssubentryitem{##2}\glsinlinesubdescformat{##4}{##5}{##6}%
    \def\gls@inlinesubsep{\glsinlinesubseparator}%
  }%
}
\compatglossarystyle{list}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \item[\glsentryitem{##1}\glstarget{##1}{##2}]
      ##3\glspostdescription\space ##5}%
  \renewcommand*\glossarysubentryfield}[6]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6.}%
}
\compatglossarystyle{listgroup}{%
  \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{listhypergroup}{%
  \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{altlist}{%

```

```

\renewcommand*\glossaryentryfield}[5]{%
  \item[\glentryitem{##1}\glstarget{##1}{##2}]%
  \mbox{}\par\nobreak\@afterheading
  ##3\glspostdescription\space ##5}%
\renewcommand*\glossarysubentryfield}[6]{%
  \par
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}##4\glspostdescription\space ##6}%
}%
\compatglossarystyle{altlistgroup}{%
  \csuse{@glscompstyle@altlist}%
}%
\compatglossarystyle{altlisthypergroup}{%
  \csuse{@glscompstyle@altlist}%
}%
\compatglossarystyle{listdotted}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \item[]\makebox[\glslistdottedwidth][l]{%
      \glentryitem{##1}\glstarget{##1}{##2}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##3}%
  \renewcommand*\glossarysubentryfield}[6]{%
    \item[]\makebox[\glslistdottedwidth][l]{%
      \glssubentryitem{##2}%
      \glstarget{##2}{##3}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##4}%
  }%
\compatglossarystyle{sublistdotted}{%
  \csuse{@glscompstyle@listdotted}%
  \renewcommand*\glossaryentryfield}[5]{%
    \item[\glentryitem{##1}\glstarget{##1}{##2}]}%
}%
\compatglossarystyle{long}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
  }%
\compatglossarystyle{longborder}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheader}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheaderborder}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{long3col}{%
  \renewcommand*\glossaryentryfield}[5]{%

```

```

\glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4 & ##6\\}%
}%
\compatglossarystyle{long3colborder}{%
\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long3colheader}{%
\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long3colheaderborder}{%
\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long4col}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
}%
\compatglossarystyle{long4colheader}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{long4colborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{long4colheaderborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4col}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colheader}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colheaderborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{longragged}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&

```



```

        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}##4\glspostdescription\space ##6%
        \tabularnewline}%
}%
\compatglossarystyle{longraggedborder}{%
  \csuse{@glscmpstyle@longragged}%
}%
\compatglossarystyle{longraggedheader}{%
  \csuse{@glscmpstyle@longragged}%
}%
\compatglossarystyle{longraggedheaderborder}{%
  \csuse{@glscmpstyle@longragged}%
}%
\compatglossarystyle{longragged3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glssubentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{longragged3colborder}{%
  \csuse{@glscmpstyle@longragged3col}%
}%
\compatglossarystyle{longragged3colheader}{%
  \csuse{@glscmpstyle@longragged3col}%
}%
\compatglossarystyle{longragged3colheaderborder}{%
  \csuse{@glscmpstyle@longragged3col}%
}%
\compatglossarystyle{altlongragged4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glssubentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%
\compatglossarystyle{altlongragged4colheader}{%
  \csuse{@glscmpstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colborder}{%
  \csuse{@glscmpstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colheaderborder}{%
  \csuse{@glscmpstyle@altlong4col}%
}%
\compatglossarystyle{index}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item\glssubentryitem{##1}\textbf{\glstarget{##1}{##2}}%

```

```

\ifx\relax##4\relax
\else
\space(##4)%
\fi
\space ##3\glspostdescription \space ##5}%
\renewcommand*\glossarysubentryfield}[6]{%
\ifcase##1\relax
% level 0
\item
\or
% level 1
\subitem
\glssubentryitem{##2}%
\else
% all other levels
\subsubitem
\fi
\textbf{\glstarget{##2}{##3}}%
\ifx\relax##5\relax
\else
\space(##5)%
\fi
\space##4\glspostdescription\space ##6}%
}%
\compatglossarystyle{indexgroup}{%
\csuse{@glscmpstyle@index}%
}%
\compatglossarystyle{indexhypergroup}{%
\csuse{@glscmpstyle@index}%
}%
\compatglossarystyle{tree}{%
\renewcommand{\glossaryentryfield}[5]{%
\hangindent0pt\relax
\parindent0pt\relax
\glssentryitem{##1}\textbf{\glstarget{##1}{##2}}%
\ifx\relax##4\relax
\else
\space(##4)%
\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\textbf{\glstarget{##2}{##3}}%
\ifx\relax##5\relax
\else
\space(##5)%

```

```

\fi
\space##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{treehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{treenoname}{%
\renewcommand{\glossaryentryfield}[5]{%
\hangindent0pt\relax
\parindent0pt\relax
\glstentryitem{##1}\textbf{\glstarget{##1}{##2}}%
\ifx\relax##4\relax
\else
\space(##4)%
\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\glstarget{##2}{\strut}%
##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treenonamegroup}{%
\csuse{@glscompstyle@treenoname}%
}%
\compatglossarystyle{treenonamehypergroup}{%
\csuse{@glscompstyle@treenoname}%
}%
\compatglossarystyle{almtree}{%
\renewcommand{\glossaryentryfield}[5]{%
\ifnum\@gls@prevlevel=0\relax
\else
\settothewidth{\glstreeindent}{\textbf{\@glswidestname\space}}%
\hangindent\glstreeindent
\parindent\glstreeindent
\fi
\makebox[Opt][r]{\makebox[\glstreeindent][l]{%
\glstentryitem{##1}\textbf{\glstarget{##1}{##2}}}}%
\ifx\relax##4\relax
\else
(##4)\space
\fi
##3\glspostdescription \space ##5\par
\def\@gls@prevlevel{0}%

```

```

}%
\renewcommand{\glossarysubentryfield}[6]{%
  \ifnum##1=1\relax
    \glosssubentryitem{##2}%
  \fi
  \ifnum\@gls@prevlevel=##1\relax
  \else
    \ifundefined{@glswidestname\romannumeral##1}{%
      \settoheight{\@gls@tmplen}{\textbf{\@glswidestname\space}}{%
      \settoheight{\@gls@tmplen}{\textbf{%
        \csname @glswidestname\romannumeral##1\endcsname\space}}}%
    \ifnum\@gls@prevlevel<##1\relax
      \setlength\glstreeindent\@gls@tmplen
      \addtolength\glstreeindent\parindent
      \parindent\glstreeindent
    \else
      \ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
        \settoheight{\glstreeindent}{\textbf{%
          \@glswidestname\space}}{%
          \settoheight{\glstreeindent}{\textbf{%
            \csname @glswidestname\romannumeral\@gls@prevlevel
              \endcsname\space}}}%
        \addtolength\parindent{-\glstreeindent}%
        \setlength\glstreeindent\parindent
      \fi
      \fi
      \hangindent\glstreeindent
      \makebox[0pt][r]{\makebox[\@gls@tmplen][l]{%
        \textbf{\glstarget{##2}{##3}}}}%
      \ifx##5\relax\relax
      \else
        (##5)\space
      \fi
      ##4\glspostdescription\space ##6\par
      \def\@gls@prevlevel{##1}%
    }%
  }%
}%
\compatglossarystyle{almtreegroup}{%
  \csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{almtreehypergroup}{%
  \csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolindex}{%
  \csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexgroup}{%
  \csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexhypergroup}{%

```

```

\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcoltree}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenoname}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenonamegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenonamehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolalmtree}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolalmtreegroup}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolalmtreehypergroup}{%
\csuse{@glscompstyle@almtree}%
}%
}%
\compatglossarystyle{superragged}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4\glspostdescription\space ##6%
\tabularnewline}%
}%
\compatglossarystyle{superraggedborder}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheader}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheaderborder}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superragged3col}{%

```

```

\renewcommand*\glossaryentryfield}[5]{%
  \glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
\renewcommand*\glossarysubentryfield}[6]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{superragged3colborder}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheader}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheaderborder}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{altsuperragged4col}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%
\compatglossarystyle{altsuperragged4colheader}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colborder}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colheaderborder}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{super}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
}%
\compatglossarystyle{superborder}{%
  \csuse{@glscompstyle@super}%
}%
\compatglossarystyle{superheader}{%
  \csuse{@glscompstyle@super}%
}%
\compatglossarystyle{superheaderborder}{%
  \csuse{@glscompstyle@super}%
}%

```

```

\compatglossarystyle{super3col}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\\}%
}%
\compatglossarystyle{super3colborder}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super3colheader}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super3colheaderborder}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super4col}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
}%
\compatglossarystyle{super4colheader}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{super4colborder}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{super4colheaderborder}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4col}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colheader}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colborder}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colheaderborder}{%
  \csuse{@glscompstyle@super4col}%
}%

```

8.6 Rollback v4.46 (glossaries-polyglossia-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-polyglossia}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \PackageWarningNoLine{glossaries}%
      {No language module detected for ‘\this@dialect’.\MessageBreak
      Language modules need to be installed separately.\MessageBreak
      Please check on CTAN for a bundle called\MessageBreak
      ‘glossaries-\CurrentTrackedLanguage’ or similar}%
    }%
  }%
}%
}
```

8.7 Rollback v4.46 (glossaries-prefix-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-prefix}[2020/03/19 v4.46 (NLCT)]
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
\ProcessOptions
\RequirePackage{glossaries}[v4.46]
\define@key{glossentry}{prefixfirst}{\def\@glo@entryprefixfirst{#1}}%
\define@key{glossentry}{prefixfirstplural}{\def\@glo@entryprefixfirstplural{#1}}%
\define@key{glossentry}{prefix}{\def\@glo@entryprefix{#1}}%
\define@key{glossentry}{prefixplural}{\def\@glo@entryprefixplural{#1}}%
\appto\@gls@keymap{,%
  {prefixfirst}{prefixfirst},%
  {prefixfirstplural}{prefixfirstplural},%
  {prefix}{prefix},%
  {prefixplural}{prefixplural}%
}
\appto\@newglossaryentryprehook{%
  \def\@glo@entryprefix{}%
  \def\@glo@entryprefixplural{}%
  \let\@glo@entryprefixfirst\@gls@default@value
```



```

\let\@glo@entryprefixfirstplural\@gls@default@value
}
\appto\@newglossaryentryposthook{%
\gls@assign@field{}\@glo@label}{prefix}\@glo@entryprefix}%
\gls@assign@field{}\@glo@label}{prefixplural}\@glo@entryprefixplural}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @prefix\endcsname}\@glo@label}{prefixfirst}%
  {\@glo@entryprefixfirst}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @prefixplural\endcsname}\@glo@label}%
  {prefixfirstplural}\@glo@entryprefixfirstplural}%
}
\newcommand*\@glsentryprefixfirst}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixfirst}}
\newcommand*\@glsentryprefixfirstplural}[1]{%
\csuse{glo@\glsdetoklabel{#1}@prefixfirstplural}}
\newcommand*\@glsentryprefix}[1]{\csuse{glo@\glsdetoklabel{#1}@prefix}}
\newcommand*\@glsentryprefixplural}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixplural}}
\newrobustcmd*\@Glsentryprefixfirst}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixfirst\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\@Glsentryprefixfirstplural}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixfirstplural\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\@Glsentryprefix}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefix\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\@Glsentryprefixplural}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixplural\endcsname}%
\xmakefirstuc\@glo@text
}
\newcommand*\@ifglshasprefix}[3]{%
\ifcsemtyp{glo@\glsdetoklabel{#1}@prefix}%
{#3}%
{#2}%
}
\newcommand*\@ifglshasprefixplural}[3]{%
\ifcsemtyp{glo@\glsdetoklabel{#1}@prefixplural}%
{#3}%
{#2}%
}
\newcommand*\@ifglshasprefixfirst}[3]{%
\ifcsemtyp{glo@\glsdetoklabel{#1}@prefixfirst}%
{#3}%
{#2}%
}
\newcommand*\@ifglshasprefixfirstplural}[3]{%
\ifcsemtyp{glo@\glsdetoklabel{#1}@prefixfirstplural}%

```

```

    {#3}%
    {#2}%
}
\providecommand{\@glsprefix@record@hook}[2]{%
  \ifdef\@glsxtr@record
    {\@glsxtr@record{#1}{#2}{glslink}}%
  {}%
}
\newcommand{\glsprefixsep}{-}
\newrobustcmd{\pgls}{\@gls@hyp@opt\@pgls}
\newcommand*{\@pgls}[2][{}]{%
  \new@ifnextchar[%
    {\@pgls@{#1}{#2}}%
    {\@pgls@{#1}{#2}[{}]}%
}
\def\@pgls@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglschasprefix{#2}{\glsentryprefix{#2}\glsprefixsep}{-}%
    }%
    {%
      \ifglschasprefixfirst{#2}{\glsentryprefixfirst{#2}\glsprefixsep}{-}%
    }%
    \@gls@{#1}{#2}[#3]%
  }%
}
\newrobustcmd{\pglsp1}{\@gls@hyp@opt\@pglsp1}
\newcommand*{\@pglsp1}[2][{}]{%
  \new@ifnextchar[%
    {\@pglsp1@{#1}{#2}}%
    {\@pglsp1@{#1}{#2}[{}]}%
}
\def\@pglsp1@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglschasprefixplural{#2}{\glsentryprefixplural{#2}\glsprefixsep}{-}%
    }%
    {%
      \ifglschasprefixfirstplural{#2}%
        {\glsentryprefixfirstplural{#2}\glsprefixsep}{-}%
    }%
    \@glspl@{#1}{#2}[#3]%
  }%
}
}

```

```

\newrobustcmd{\Pgl}{\@gls@hyp@opt\Pgl}
\newcommand*{\@Pgl}[2][\@Pgl]{}
\new@ifnextchar[
{\@Pgl@{#1}{#2}}%
{\@Pgl@{#1}{#2} []}%
}
\def\@Pgl@#1#2[#3]{%
\@glsprefix@record@hook{#1}{#2}%
\glsdoifexists{#2}%
{%
\ifglsused{#2}%
{%
\ifglshasprefix{#2}%
{%
\Glsentryprefix{#2}%
\glsprefixsep
\@gls@{#1}{#2}[#3]%
}%
{\@Gls@{#1}{#2}[#3]}%
}%
{%
\ifglshasprefixfirst{#2}%
{%
\Glsentryprefixfirst{#2}%
\glsprefixsep
\@gls@{#1}{#2}[#3]%
}%
{\@Gls@{#1}{#2}[#3]}%
}%
}%
}
\newrobustcmd{\Pglsp}{\@gls@hyp@opt\Pglsp}
\newcommand*{\@Pglsp}[2][\@Pglsp]{}
\new@ifnextchar[
{\@Pglsp@{#1}{#2}}%
{\@Pglsp@{#1}{#2} []}%
}
\def\@Pglsp@#1#2[#3]{%
\@glsprefix@record@hook{#1}{#2}%
\glsdoifexists{#2}%
{%
\ifglsused{#2}%
{%
\ifglshasprefixplural{#2}%
{%
\Glsentryprefixplural{#2}%
\glsprefixsep
\@glspl@{#1}{#2}[#3]%
}%
{\@Glspl@{#1}{#2}[#3]}%
}%
}
}

```

```

}%
{%
  \ifglshasprefixfirstplural{#2}%
  {%
    \Glsentryprefixfirstplural{#2}%
    \glsprefixsep
    \@glspl@{#1}{#2}[#3]%
  }%
  {\@Glspl@{#1}{#2}[#3]}%
}%
}%
}
\newrobustcmd{\PGLS}{\@gls@hyp@opt\PGLS}
\newcommand*{\@PGLS}[2][ ]{%
  \new@ifnextchar [%
    {\@PGLS@{#1}{#2}}%
    {\@PGLS@{#1}{#2} [ ]}%
  }
}
\def\@PGLS@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglshasprefix{#2}%
      {\mfirstucMakeUppercase{\glsentryprefix{#2}\glsprefixsep}}{%
    }%
    {%
      \ifglshasprefixfirst{#2}%
      {\mfirstucMakeUppercase{\glsentryprefixfirst{#2}\glsprefixsep}}{%
    }%
    \@GLS@{#1}{#2}[#3]%
  }%
}
}
\newrobustcmd{\PGLSpl}{\@gls@hyp@opt\PGLSpl}
\newcommand*{\@PGLSpl}[2][ ]{%
  \new@ifnextchar [%
    {\@PGLSpl@{#1}{#2}}%
    {\@PGLSpl@{#1}{#2} [ ]}%
  }
}
\def\@PGLSpl@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglshasprefixplural{#2}%
      {\mfirstucMakeUppercase{\glsentryprefixplural{#2}\glsprefixsep}}{%
    }%
    {%

```

```

        \ifglshasprefixfirstplural{#2}%
        {\mfirstucMakeUppercase{\glsentryprefixfirstplural{#2}\glsprefixsep}}{}%
    }%
    \@GLSpl@{#1}{#2}{#3}%
} %
}

```

8.8 Rollback v4.46 (glossary-hypernav-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-hypernav}[2020/03/19 v4.46 (NLCT)]
\newcommand*{\glsnavhyperlink}[3][\@glo@type]{%
  \edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%
  \glslink{\glsnavhyperlinkname{#1}{#2}}{#3}}
\newcommand*{\glsnavhyperlinkname}[2]{\glsn:#1@#2}
\newcommand*{\glsnavhypertarget}[3][\@glo@type]{%
  \glsnavhypertarget{#1}{#2}{#3}%
}
\newcommand*{\@glsnavhypertarget}[3]{%
  \protected@write\@auxout{}{\string\@gls@hypergroup{#1}{#2}}%
  \gls@target{\glsnavhyperlinkname{#1}{#2}}{#3}%
  \expandafter\let
    \expandafter\@gls@list\csname @gls@hypergroup@list@#1\endcsname
  \@for\@gls@elem:=\@gls@list\do{%
    \ifthenelse{\equal{\@gls@elem}{#2}}{\@endfortrue}{}%
  }
  \if@endfor
  \else
    \GlossariesWarningNoLine{Navigation panel
      for glossary type ‘#1’^^Jmissing group ‘#2’}%
    \gdef\gls@hypergroup@prerun{%
      \GlossariesWarningNoLine{Navigation panel
        has changed. Rerun LaTeX}}%
    \fi
  }
\let\gls@hypergroup@prerun\relax
\AtEndDocument{\gls@hypergroup@prerun}
\newcommand*{\@gls@hypergroup}[2]{%
\@ifundefined{@gls@hypergroup@list@#1}{%
  \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{#2}%
}{%
  \expandafter\let\expandafter\@gls@tmp
    \csname @gls@hypergroup@list@#1\endcsname
  \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{%
    \@gls@tmp,#2}%
}%
}
\newcommand*{\glsnavigation}{%
  \def\@gls@between{}%
  \ifcsundef{@gls@hypergroup@list@\@glo@type}%

```

```

{%
  \def\@gls@list{%
}%
{%
  \expandafter\let\expandafter\@gls@list
  \csname @gls@hypergroup@list@\@glo@type\endcsname
}%
\@for\@gls@tmp:=\@gls@list\do{%
  \@gls@between
  \@gls@getgrouptitle{\@gls@tmp}{\@gls@grptitle}%
  \glsnavhyperlink{\@gls@tmp}{\@gls@grptitle}%
  \let\@gls@between\gls@hypernavsep
}%
}
\newcommand*\gls@hypernavsep{\space\textbar\space}
\newcommand*\gls@symbolnav{%
  \glsnavhyperlink{gls@symbols}{\gls@getgrouptitle{gls@symbols}}%
  \gls@hypernavsep
  \glsnavhyperlink{gls@numbers}{\gls@getgrouptitle{gls@numbers}}%
  \gls@hypernavsep
}

```

8.9 Rollback v4.46 (glossary-inline-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-inline}[2020/03/19 v4.46 (NLCT)]
\newglossarystyle{inline}{%
  \renewenvironment{theglossary}%
  {%
    \def\@gls@inlinesep{}%
    \def\@gls@inlinesubsep{}%
    \def\@gls@inlinepostchild{}%
  }%
  {\gls@postinline}%
\renewcommand*\@glossaryheader{}%
\renewcommand*\@gls@groupheading[1]{}%
\renewcommand*\@glossentry[2]{%
  \gls@inlinedopostchild
  \gls@inlinesep
  \gls@entryitem{##1}%
  \gls@inlinenameformat{##1}%
  \glossentryname{##1}%
}%
\ifglsdescsuppressed{##1}%
{%
  \gls@inlineemptydescformat
  {%
    \glossentrysymbol{##1}%
  }%
}

```

```

    {%
      ##2%
    }%
  }%
  {%
    \ifglshasdesc{##1}%
    {\glsinlinedescformat{\glossentrydesc{##1}}{\glossentrysymbol{##1}}{##2}}%
    {\glsinlineemptydescformat{\glossentrysymbol{##1}}{##2}}%
  }%
  \ifglshaschildren{##1}%
  {%
    \glsresetsubentrycounter
    \glsinlineparentchildseparator
    \def\gls@inlinesubsep{}%
    \def\gls@inlinepostchild{\glsinlinepostchild}%
  }%
  {}%
  \def\gls@inlinesep{\glsinlineseparator}%
}%
\renewcommand{\subglossentry}[3]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{%
    \glossentryname{##2}}%
  \glsesubentryitem{##2}%
  \glsinlinesubdescformat{\glossentrydesc{##2}}{\glossentrysymbol{##2}}{##3}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%
\renewcommand*{\glsgroupskip}{}%
}
\newcommand*{\glsinlinedopostchild}{%
  \gls@inlinepostchild
  \def\gls@inlinepostchild{}%
}
\newcommand*{\glsinlineseparator}{;\space}
\newcommand*{\glsinlinesubseparator}{,\space}
\newcommand*{\glsinlineparentchildseparator}{:\space}
\newcommand*{\glsinlinepostchild}{}
\newcommand*{\glspostinline}{\glspostdescription\space}
\newcommand*{\glsinlinenameformat}[2]{\glstarget{#1}{#2}}
\newcommand*{\glsinlinedescformat}[3]{\space#1}
\newcommand*{\glsinlineemptydescformat}[2]{}
\newcommand*{\glsinlinesubnameformat}[2]{\glstarget{#1}{}}
\newcommand*{\glsinlinesubdescformat}[3]{#1}

```

8.10 Rollback v4.46 (glossary-list-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-list}[2020/03/19 v4.46 (NLCT)]
\providecommand{\indexspace}{%

```

```

\par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*\glslistgroupheaderfmt}[1]{#1}
\newcommand*\glslistnavigationitem}[1]{\item[#1]}
\newglossarystyle{list}{%
  \renewenvironment{theglossary}%
    {\begin{description}}{\end{description}}%
  \renewcommand*\glossaryheader{}%
  \renewcommand*\glsgroupheading}[1]{}%
  \renewcommand*\glossentry}[2]{%
    \item[\glsentryitem{##1}]%
      \glstarget{##1}{\glossentryname{##1}}%
      \glossentrydesc{##1}\glspostdescription\space ##2}%
  \renewcommand*\subglossentry}[3]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\space
    \glossentrydesc{##2}\glspostdescription\space ##3.}%
  \renewcommand*\glsgroupskip{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{listgroup}{%
  \setglossarystyle{list}%
  \renewcommand*\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
\newglossarystyle{listhypergroup}{%
  \setglossarystyle{list}%
  \renewcommand*\glossaryheader{%
    \glslistnavigationitem{\glsnavigation}}%
  \renewcommand*\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}]}]}
\newglossarystyle{altlist}{%
  \setglossarystyle{list}%
  \renewcommand*\glossentry}[2]{%
    \item[\glsentryitem{##1}]%
      \glstarget{##1}{\glossentryname{##1}}%
      \mbox{}\par\nobreak\@afterheading
      \glossentrydesc{##1}\glspostdescription\space ##2}%
  \renewcommand*\subglossentry}[3]{%
    \par
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space ##3}%
}
\newglossarystyle{altlistgroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
\newglossarystyle{altlisthypergroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*\glossaryheader{%
    \glslistnavigationitem{\glsnavigation}}%
}

```



```

\renewcommand*{\glsgroupheading}[1]{%
  \item[\glslistgroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}]%
\newglossarystyle{listdotted}{%
  \setglossarystyle{list}%
  \renewcommand*{\glossentry}[2]{%
    \item[]\makebox[\glslistdottedwidth][1]{%
      \glstarget{##1}{\glossentryname{##1}}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut\glossentrydesc{##1}}%
  \renewcommand*{\subglossentry}[3]{%
    \item[]\makebox[\glslistdottedwidth][1]{%
      \glssubentryitem{##2}%
      \glstarget{##2}{\glossentryname{##2}}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut\glossentrydesc{##2}}%
  }
\newlength\glslistdottedwidth
\setlength{\glslistdottedwidth}{.5\hspace}
\newglossarystyle{sublistdotted}{%
  \setglossarystyle{listdotted}%
  \renewcommand*{\glossentry}[2]{%
    \item[\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}}]}%
}

```

8.11 Rollback v4.46 (glossary-list-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-long}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
  \newlength\glsdescwidth
  \setlength{\glsdescwidth}{0.6\hspace}
}{%
\@ifundefined{glspagelistwidth}{%
  \newlength\glspagelistwidth
  \setlength{\glspagelistwidth}{0.1\hspace}
}{%
\newglossarystyle{long}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}[lp{\glsdescwidth}]}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand*{\glossentry}[2]{%
    \glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand*{\subglossentry}[3]{%
    &

```

```

        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
        ##3\tabularnewline
    }%
    \ifglsgroupskip
        \renewcommand*\glsgroupskip{ }%
    \else
        \renewcommand*\glsgroupskip{ & \tabularnewline}%
    \fi
}
\newglossarystyle{longborder}{%
    \setglossarystyle{long}%
    \renewenvironment{theglossary}{%
        \begin{longtable}{|l|p{\glsdescwidth}|}{\end{longtable}}%
        \renewcommand*\glossaryheader{\hline\endhead\hline\endfoot}%
    }
    \newglossarystyle{longheader}{%
        \setglossarystyle{long}%
        \renewcommand*\glossaryheader{%
            \bfseries \entryname & \bfseries \descriptionname\tabularnewline\endhead}%
        }
    \newglossarystyle{longheaderborder}{%
        \setglossarystyle{longborder}%
        \renewcommand*\glossaryheader{%
            \hline\bfseries \entryname & \bfseries
            \descriptionname\tabularnewline\hline
            \endhead
            \hline\endfoot}%
        }
    \newglossarystyle{long3col}{%
        \renewenvironment{theglossary}{%
            \begin{longtable}{|lp{\glsdescwidth}p{\glspagelistwidth}|}{%
                \end{longtable}}%
            \renewcommand*\glossaryheader{ }%
            \renewcommand*\glsgroupheading}[1]{ }%
            \renewcommand{\glossentry}[2]{%
                \glssubentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
                \glossentrydesc{##1} & ##2\tabularnewline
            }%
            \renewcommand{\subglossentry}[3]{%
                &
                \glssubentryitem{##2}%
                \glstarget{##2}{\strut}\glossentrydesc{##2} &
                ##3\tabularnewline
            }%
            \ifglsgroupskip
                \renewcommand*\glsgroupskip{ }%
            \else
                \renewcommand*\glsgroupskip{ & & \tabularnewline}%
            \fi

```

```

}
\newglossarystyle{long3colborder}{%
  \setglossarystyle{long3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long3colheader}{%
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long3colheaderborder}{%
  \setglossarystyle{long3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}
}
\newglossarystyle{long4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l|l|l|l}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{long4colheader}{%
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&

```

```

        \bfseries \symbolname&
        \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long4colborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long4colheaderborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}
\newglossarystyle{altlong4col}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheader}{%
  \setglossarystyle{long4colheader}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colborder}{%
  \setglossarystyle{long4colborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheaderborder}{%
  \setglossarystyle{long4colheaderborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%
    {\end{longtable}}%
}
}

```

8.12 Rollback v4.46 (glossary-longbooktabs-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-longbooktabs}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{booktabs}
\RequirePackage{glossary-long}[=v4.46]
\RequirePackage{glossary-longragged}[=v4.46]
\newglossarystyle{long-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname & \bfseries
    \descriptionname\tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{long3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname &
    \bfseries \descriptionname &
    \bfseries \pagelistname
    \tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{long4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname &
    \bfseries \descriptionname &
    \bfseries \symbolname &
    \bfseries \pagelistname
    \tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{altlong4col-booktabs}{%
  \glspatchLToutput

```

```

\setglossarystyle{long4col-booktabs}%
\renewenvironment{theglossary}%
  {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
  {\end{longtable}}%
}
\newglossarystyle{longragged-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{longragged3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlongragged4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{altlong4col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newcommand*{\glsLTpenaltycheck}{%
  \ifnum\outputpenalty=-50\vskip-\normalbaselineskip\relax\fi
}
\newcommand{\glspenaltygroupskip}{%
  \noalign{\penalty-50\vskip\normalbaselineskip}}
\let\@gls@org@LT@output\LT@output
\newcommand*{\glsrestoreLToutput}{\let\LT@output\@gls@org@LT@output}
\newcommand*{\glspatchLToutput}{%
  \renewcommand*{\LT@output}{%
    \ifnum\outputpenalty <-\@Mi
      \ifnum\outputpenalty > -\LT@end@pen
        \LT@err{floats and marginpars not allowed in a longtable}\@ehc
      \else
        \setbox\z@\vbox{\unvbox\@cclv}%
        \ifdim \ht\LT@lastfoot>\ht\LT@foot
          \dimen@pagegoal
          \advance\dimen@-\ht\LT@lastfoot
          \ifdim\dimen@<\ht\z@
            \setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%
            \@makecol
            \@outputpage
            \setbox\z@\vbox{\box\LT@head\glsLTpenaltycheck}%
          \fi
        \fi
      \fi
    }%
}

```

```

        \fi
        \fi
        \global\@colroom\@colht
        \global\vsizel\@colht
        {\unvbox\z@\box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%
        \fi
    \else
        \setbox\@cclv\vbox{\unvbox\@cclv\copy\LT@foot\vss}%
        \@makecol
        \@outputpage
        \global\vsizel\@colroom
        \copy\LT@head
        \glsLTpenaltycheck
        \nobreak
    \fi
}
}
}

```

8.13 Rollback v4.46 (glossary-longragged-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-longragged}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{array}
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
    \newlength\glsdescwidth
    \setlength\glsdescwidth{0.6\hsize}
}{%
\@ifundefined{glspagelistwidth}{%
    \newlength\glspagelistwidth
    \setlength\glspagelistwidth{0.1\hsize}
}{%
\newglossarystyle{longragged}{%
    \renewenvironment{theglossary}{%
        {\begin{longtable}{1>\raggedright}p{glsdescwidth}}}%
        {\end{longtable}}}%
    \renewcommand*\glossaryheader{}%
    \renewcommand*\glsgroupheading}[1]{}%
    \renewcommand\glossentry}[2]{%
        \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1}\glspostdescription\space ##2%
        \tabularnewline
    }%
    \renewcommand\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2}%
        \glspostdescription\space ##3%
        \tabularnewline
    }%
}
}
}

```

```

}%
\ifglsgroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip{ & \tabularnewline}%
\fi
}
\newglossarystyle{longraggedborder}{%
  \setglossarystyle{longragged}%
  \renewenvironment{theglossary}{%
    \begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}||}%
    {\end{longtable}}%
  \renewcommand*\glossaryheader{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{longraggedheader}{%
  \setglossarystyle{longragged}%
  \renewcommand*\glossaryheader{%
    \bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\endhead}%
}
\newglossarystyle{longraggedheaderborder}{%
  \setglossarystyle{longraggedborder}%
  \renewcommand*\glossaryheader{%
    \hline\bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\hline
    \endhead
    \hline\endfoot}%
}
\newglossarystyle{longragged3col}{%
  \renewenvironment{theglossary}%
  {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}%
    >{\raggedright}p{\glspagelistwidth}}%
  {\end{longtable}}%
  \renewcommand*\glossaryheader{}%
  \renewcommand*\glsgroupeheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    ##3\tabularnewline
  }%
\ifglsgroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip{ & & \tabularnewline}%
\fi

```



```

}
\newglossarystyle{longragged3colborder}{%
  \setglossarystyle{longragged3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{longragged3colheader}{%
  \setglossarystyle{longragged3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{longragged3colheaderborder}{%
  \setglossarystyle{longragged3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}
\newglossarystyle{altlongragged4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{altlongragged4colheader}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%

```

```

        {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
          >{\raggedright}p{\glspagelistwidth}}}%
      {\end{longtable}}%
\renewcommand*{\glossaryheader}{%
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{altlongragged4colborder}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|l%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{altlongragged4colheaderborder}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|l%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}

```

8.14 Rollback v4.46 (glossary-mcols-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-mcols}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{multicol}
\RequirePackage{glossary-tree}[=v4.46]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\gls mcols}{2}
\newglossarystyle{mcolindex}{%
  \setglossarystyle{index}%
  \renewenvironment{theglossary}%
    {%
      \begin{multicols}{\gls mcols}
      \setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%
      \let\item\glstreeitem
      \let\subitem\glstreesubitem
      \let\subsubitem\glstreesubsubitem
    }
  }

```

```

    }%
    {\end{multicols}}%
}
\newglossarystyle{mcolindexgroup}{%
\setglossarystyle{mcolindex}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\indexspace}%
}
\newglossarystyle{mcolindexhypergroup}{%
\setglossarystyle{mcolindex}%
\renewcommand*{\glossaryheader}{%
\item\glstreenavigationfmt{\glsnavigation}\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
\indexspace}%
}
\newglossarystyle{mcolindexspannav}{%
\setglossarystyle{index}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glscols}{\noindent\glstreenavigationfmt{\glsnavigation}}
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
\let\item\glstreeitem}%
{\end{multicols}}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
\indexspace}%
}
}
\newglossarystyle{mcoltree}{%
\setglossarystyle{tree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glscols}
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%
}
\newglossarystyle{mcoltreegroup}{%
\setglossarystyle{mcoltree}%
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreehypergroup}{%
\setglossarystyle{mcoltree}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
}

```

```

\renewcommand*{\glsgroupheading}[1]{%
  \par\noindent
  \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}\par
  \indexspace}%
}
\newglossarystyle{mcoltreesspannav}{%
  \setglossarystyle{tree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}\par
    \indexspace}%
  }
}
\newglossarystyle{mcoltreenoname}{%
  \setglossarystyle{treenoname}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}
\newglossarystyle{mcoltreenonamegroup}{%
  \setglossarystyle{mcoltreenoname}%
  \renewcommand*{\glsgroupheading}[1]{\par
  \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreenonamehypergroup}{%
  \setglossarystyle{mcoltreenoname}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}\par
    \indexspace}%
  }
}
\newglossarystyle{mcoltreenonamespannav}{%
  \setglossarystyle{treenoname}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }
}

```

```

}%
{\end{multicols}}%
\renewcommand*{\glsgroupheading}[1]{%
  \par\noindent
  \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}
\newglossarystyle{mcolalmtree}{%
  \setglossarystyle{almtree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glscols}
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
}
\newglossarystyle{mcolalmtreegroup}{%
  \setglossarystyle{mcolalmtree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcolalmtreehypergroup}{%
  \setglossarystyle{mcolalmtree}%
  \renewcommand*{\glossaryheader}{%
    \par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcolalmtreespannav}{%
  \setglossarystyle{almtree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glscols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
}

```

```

\renewcommand*{\glsgroupheading}[1]{%
  \par
  \def\@gls@prevlevel{-1}%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}

```

8.15 Rollback v4.46 (glossary-super-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-super}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
  \newlength\glsdescwidth
  \setlength{\glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength\glspagelistwidth
  \setlength{\glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{super}{%
  \renewenvironment{theglossary}%
    {\tablehead{}}\tabletail{}}%
  \begin{supertabular}{lp{\glsdescwidth}}%
  {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glstryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}}%

```

```

        \begin{supertabular}{|l|p{\glsdescwidth}|}%
        {\end{supertabular}}%
    }
    \newglossarystyle{superheader}{%
        \setglossarystyle{super}%
    \renewenvironment{theglossary}%
        {\tablehead{\bfseries \entryname &
            \bfseries\descriptionname\tabularnewline}%
            \tabletail{}}%
        {\begin{supertabular}{|l|p{\glsdescwidth}}}%
        {\end{supertabular}}%
    }
    \newglossarystyle{superheaderborder}{%
        \setglossarystyle{super}%
    \renewenvironment{theglossary}%
        {\tablehead{\hline\bfseries \entryname &
            \bfseries \descriptionname\tabularnewline\hline}%
            \tabletail{\hline}}
        {\begin{supertabular}{|l|p{\glsdescwidth}|}%
        {\end{supertabular}}%
    }
    \newglossarystyle{super3col}{%
    \renewenvironment{theglossary}%
        {\tablehead{} \tabletail{}}%
        {\begin{supertabular}{|lp{\glsdescwidth}p{\glspagelistwidth}}}%
        {\end{supertabular}}%
    \renewcommand*\glossaryheader{}%
    \renewcommand*\glsgroupheading}[1]{}%
    \renewcommand\glossentry[2]{%
        \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1} & ##2\tabularnewline
    }%
    \renewcommand\subglossentry[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2} &
        ##3\tabularnewline
    }%
    \ifglsnogroupskip
        \renewcommand*\glsgroupskip{}%
    \else
        \renewcommand*\glsgroupskip}{& & \tabularnewline}%
    \fi
    }
    \newglossarystyle{super3colborder}{%
        \setglossarystyle{super3col}%
    \renewenvironment{theglossary}%
        {\tablehead{\hline}\tabletail{\hline}%
            \begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
            {\end{supertabular}}%

```

```

}
\newglossarystyle{super3colheader}{%
  \setglossarystyle{super3col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    {\begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{super3colheaderborder}{%
  \setglossarystyle{super3colborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname&\bfseries\descriptionname&
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    {\begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
    {\end{supertabular}}}%
}
\newglossarystyle{super4col}{%
  \renewenvironment{theglossary}%
    {\tablehead{}}\tabletail{}}%
    {\begin{supertabular}{llll}}{%
    \end{supertabular}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
  \fi
}
\newglossarystyle{super4colheader}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}%
    \tabletail{}}%

```



```

        \begin{supertabular}{|l|l|l|}%
        \end{supertabular}}%
}
\newglossarystyle{super4colborder}{%
\setglossarystyle{super4col}%
\renewenvironment{theglossary}%
{\tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}{|l|l|l|l|l|}%
\end{supertabular}}%
}
\newglossarystyle{super4colheaderborder}{%
\setglossarystyle{super4col}%
\renewenvironment{theglossary}%
{\tablehead{\hline\bfseries\entryname&\bfseries\descriptionname&
\bfseries\symbolname &
\bfseries\pagelistname\tabularnewline\hline}%
\tabletail{\hline}%
\begin{supertabular}{|l|l|l|l|l|}%
\end{supertabular}}%
}
\newglossarystyle{altsuper4col}{%
\setglossarystyle{super4col}%
\renewenvironment{theglossary}%
{\tablehead{} \tabletail{}}%
\begin{supertabular}{|lp{\glsdescwidth}lp{\glspagelistwidth}}%
\end{supertabular}}%
}
\newglossarystyle{altsuper4colheader}{%
\setglossarystyle{super4colheader}%
\renewenvironment{theglossary}%
{\tablehead{\bfseries\entryname&\bfseries\descriptionname&
\bfseries\symbolname &
\bfseries\pagelistname\tabularnewline}\tabletail{}}%
\begin{supertabular}{|lp{\glsdescwidth}lp{\glspagelistwidth}}%
\end{supertabular}}%
}
\newglossarystyle{altsuper4colborder}{%
\setglossarystyle{super4colborder}%
\renewenvironment{theglossary}%
{\tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}%
{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}%
\end{supertabular}}%
}
\newglossarystyle{altsuper4colheaderborder}{%
\setglossarystyle{super4colheaderborder}%
\renewenvironment{theglossary}%
{\tablehead{\hline
\bfseries\entryname &
\bfseries\descriptionname &

```

```

        \bfseries\symbolname &
        \bfseries\pagelistname\tabularnewline\hline}%
\tabletail{\hline}%
\begin{supertabular}%
  {1|p{\glsdescwidth}|1|p{\glspagelistwidth}|}%
\end{supertabular}}%
}

```

8.16 Rollback v4.46 (glossary-superragged-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-superragged}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{array}
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
  \newlength\glsdescwidth
  \setlength\glsdescwidth{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength\glspagelistwidth
  \setlength\glspagelistwidth{0.1\hsize}
}{%}
\newglossarystyle{superragged}{%
  \renewenvironment{theglossary}%
    {\tablehead{\tabletail}}%
    {\begin{supertabular}{1>{\raggedright}p{\glsdescwidth}}}%
    {\end{supertabular}}%
  \renewcommand*\glossaryheader{}%
  \renewcommand*\glsgroupheading}[1]{%}
  \renewcommand\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2%
    \tabularnewline
  }%
  \renewcommand\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3%
    \tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*\glsgroupskip}{%}
  \else
    \renewcommand*\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superraggedborder}{%
  \setglossarystyle{superragged}%
}

```

```

\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
   \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|}}%
  {\end{supertabular}}%
}
\newglossarystyle{superraggedheader}{%
  \setglossarystyle{superragged}%
\renewenvironment{theglossary}%
  {\tablehead{\bfseries \entryname & \bfseries \descriptionname
  \tabularnewline}%
  \tabletail{}}%
  \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}}}%
  {\end{supertabular}}%
}
\newglossarystyle{superraggedheaderborder}{%
  \setglossarystyle{superragged}%
\renewenvironment{theglossary}%
  {\tablehead{\hline\bfseries \entryname &
  \bfseries \descriptionname\tabularnewline\hline}%
  \tabletail{\hline}%
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|}}%
  {\end{supertabular}}%
}
\newglossarystyle{superragged3col}{%
\renewenvironment{theglossary}%
  {\tablehead{} \tabletail{}%
  \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}%
  >{\raggedright}p{\glspagelistwidth}}}%
  {\end{supertabular}}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  ##2\tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  ##3\tabularnewline
}%
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& \tabularnewline}%
\fi
}
\newglossarystyle{superragged3colborder}{%
  \setglossarystyle{superragged3col}%

```

```

\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
   \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
    >{\raggedright}p{\glspagelistwidth}|}%
   {\end{supertabular}}}%
}
\newglossarystyle{superragged3colheader}{%
  \setglossarystyle{superragged3col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
     \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    \begin{supertabular}{|l>{\raggedright}p{\glsdescwidth}%
     >{\raggedright}p{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{superragged3colheaderborder}{%
  \setglossarystyle{superragged3colborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
     \bfseries\entryname&\bfseries\descriptionname&
     \bfseries\pagelistname\tabularnewline\hline}%
     \tabletail{\hline}%
     \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
      >{\raggedright}p{\glspagelistwidth}|}%
     {\end{supertabular}}}%
}
\newglossarystyle{altsuperragged4col}{%
  \renewenvironment{theglossary}%
    {\tablehead{} \tabletail{}%
     \begin{supertabular}{|l>{\raggedright}p{\glsdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}}}%
     {\end{supertabular}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
  \fi

```

```

}
\newglossarystyle{altsuperragged4colheader}{%
  \setglossarystyle{altsuperragged4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    {\begin{supertabular}{l>{\raggedright}p{\glsdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuperragged4colborder}{%
  \setglossarystyle{altsuper4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}%
      {l|>{\raggedright}p{\glsdescwidth}l|}%
      >{\raggedright}p{\glspagelistwidth}l}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuperragged4colheaderborder}{%
  \setglossarystyle{altsuperragged4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname &
      \bfseries\descriptionname &
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}%
      {l|>{\raggedright}p{\glsdescwidth}l|}%
      >{\raggedright}p{\glspagelistwidth}l}}%
    {\end{supertabular}}}%
}

```

8.17 Rollback v4.46 (glossary-tree-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-tree}[2020/03/19 v4.46 (NLCT)]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glstreenamefmt}[1]{\textbf{#1}}
\newcommand*{\glstreegroupheaderfmt}[1]{\glstreenamefmt{#1}}
\newcommand*{\glstreenavigationfmt}[1]{\glstreenamefmt{#1}}
\ifdef\@idxitem
{\newcommand{\glstreeitem}{\@idxitem}}
{\newcommand{\glstreeitem}{\par\hangindent40\p@}}
\ifdef\subitem

```

```

{\let\glstreesubitem\subitem}
{\newcommand\glstreesubitem{\glstreeitem\hspace*{20\p@}}
\ifdef\subsubitem
{\let\glstreesubsubitem\subsubitem}
{\newcommand\glstreesubsubitem{\glstreeitem\hspace*{30\p@}}
\newcommand{\glstreepredesc}{\space}
\newcommand{\glstreechildpredesc}{\space}
\newglossarystyle{index}{%
\renewenvironment{theglossary}%
{\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
\let\item\glstreeitem
\let\subitem\glstreesubitem
\let\subsubitem\glstreesubsubitem
}%
{\par}%
\renewcommand*\{glossaryheader}{}%
\renewcommand*\{glsgroupheading}[1]{}%
\renewcommand*\{glossentry}[2]{%
\item\glstreeitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
\ifglshassymbol{##1}{\space\glossentrysymbol{##1}}{}}%
\glstreepredesc \glossentrydesc{##1}\glspostdescription\space ##2%
}%
\renewcommand{\subglossentry}[3]{%
\ifcase##1\relax
% level 0
\item
\or
% level 1
\subitem
\glssubentryitem{##2}%
\else
% all other levels
\subsubitem
\fi
\glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
\ifglshassymbol{##2}{\space\glossentrysymbol{##2}}{}}%
\glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3%
}%
\renewcommand*\{glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
\newglossarystyle{indexgroup}{%
\setglossarystyle{index}%
\renewcommand*\{glsgroupheading}[1]{%
\item\glstreegroupheaderfmt{\glsgrouptitle{##1}}%
\indexspace
}%
}
\newglossarystyle{indexhypergroup}{%
\setglossarystyle{index}%
\renewcommand*\{glossaryheader}{%

```

```

\item\glstreenavigationfmt{\glsnavigation}\indexspace}%
\renewcommand*\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
\indexspace}%
}
\newglossarystyle{tree}{%
\renewenvironment{theglossary}%
{\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}}%
{}%
\renewcommand*\glossaryheader}{}%
\renewcommand*\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\hangindent0pt\relax
\parindent0pt\relax
\glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
\ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
\glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
}%
\renewcommand{\subglossentry}[3]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
\ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
\glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3\par
}%
\renewcommand*\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
\newglossarystyle{treegroup}{%
\setglossarystyle{tree}%
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par
\indexspace}%
}
\newglossarystyle{treehypergroup}{%
\setglossarystyle{tree}%
\renewcommand*\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
\newlength\glstreeindent
\setlength{\glstreeindent}{10pt}
\newglossarystyle{treenoname}{%

```

```

\renewenvironment{theglossary}%
  {\setlength{\parindent}{0pt}%
   \setlength{\parskip}{0pt plus 0.3pt}}%
  {}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
  \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
  \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
}%
\renewcommand{\subglossentry}[3]{%
  \hangindent##1\glstreeindent\relax
  \parindent##1\glstreeindent\relax
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \glstarget{##2}{\strut}%
  \glossentrydesc{##2}\glspostdescription\space##3\par
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{treenonamegroup}{%
  \setglossarystyle{treenoname}%
  \renewcommand{\glsgroupheading}[1]{\par
  \noindent\glstreegroupheaderfmt
  {\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{treenonamehypergroup}{%
  \setglossarystyle{treenoname}%
  \renewcommand*{\glossaryheader}{%
  \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
  \par\noindent
  \glstreegroupheaderfmt
  {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}
\newrobustcmd*{\glsfindwidesttoplevelname}[1][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
  \forglsentries[\@gls@type]{\@glo@label}%
  {%
  \ifglshasparent{\@glo@label}%
  {}%
  {%

```



```

\addtolength\glstreeindent\parindent
\parindent\glstreeindent
\else
\@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
\settowidth{\glstreeindent}{\glstreenamefmt{%
\@glswidestname\space}}}{%
\settowidth{\glstreeindent}{\glstreenamefmt{%
\csname @glswidestname\romannumeral\@gls@prevlevel
\endcsname\space}}}{%
\addtolength\parindent{-\glstreeindent}%
\setlength\glstreeindent\parindent
\fi
\fi
\hangindent\glstreeindent
\makebox[Opt][r]{\glstreenamebox{\gls@tmplen}{%
\glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}{%
\ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
\glossentrydesc{##2}\glspostdescription\space ##3\par
\def\@gls@prevlevel{##1}%
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{almtreegroup}{%
\setglossarystyle{almtree}%
\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
\par\indexspace}%
}
\newglossarystyle{almtreehypergroup}{%
\setglossarystyle{almtree}%
\renewcommand*{\glossaryheader}{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}}

```

8.18 Rollback v4.49 (glossaries-2021-11-01.sty)

```

\NeedsTeXFormat{LaTeX2e}
\DeclareRelease{v4.46}{2020-03-19}{glossaries-2020-03-19.sty}
\DeclareCurrentRelease{v4.49}{2021-11-01}
\ProvidesPackage{glossaries}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{ifthen}
\RequirePackage{xkeyval}[2006/11/18]
\RequirePackage{mfirstuc}
\RequirePackage{textcase}
\renewcommand*{\mfirstucMakeUppercase}{\MakeTextUppercase}%
\RequirePackage{xfor}
\RequirePackage{datatool-base}
\RequirePackage{amsgen}
\RequirePackage{etoolbox}
\newif\if@gls@docloaded
\@ifpackageloaded{doc}{%
  {%
    \@gls@docloadedtrue
  }%
  {%
    \@ifclassloaded{nlctdoc}{\@gls@docloadedtrue}{\@gls@docloadedfalse}%
  }
}
\if@gls@docloaded
  \let\glsorg@theglossary\theglossary
  \let\glsorg@endtheglossary\endtheglossary
  \let\glsorg@PrintChanges\PrintChanges
  \renewcommand{\PrintChanges}{%
    \begingroup
      \let\theglossary\glsorg@theglossary
      \let\endtheglossary\glsorg@endtheglossary
      \glsorg@PrintChanges
    \endgroup
  }
\fi
\newif\if@gls@debug
\define@choicekey{glossaries.sty}{debug}[\gls@debug@val\gls@debug@nr]{%
  {true,false,showtargets,showaccsupp}[true]{%
    \ifcase\gls@debug@nr\relax
      % debug=true
      \@gls@debugtrue
      \renewcommand*{\GlossariesWarning}[1]{%
        \PackageWarning{glossaries}{##1}%
      }%
      \renewcommand*{\GlossariesWarningNoLine}[1]{%
        \PackageWarningNoLine{glossaries}{##1}%
      }%
      \let\gls@showtarget\@gobble
      \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
    \or

```

```

% debug=false
\@gls@debugfalse
\let\@glsshowtarget\@gobble
\let\@glsshowaccsupp\@gobblethree
\PackageInfo{glossaries}{debug mode OFF}%
\or
% debug=showtargets
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowtarget}{\@@glsshowtarget}%
\or
% debug=showaccsupp
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
\fi
}
\newcommand*\glsshowtarget}[1]{%
\ifmmode
  \nfss@text{\glsshowtargetfont [#1]}%
\else
  \ifinner
    {\glsshowtargetfont [#1]}%
  \else
    \glsshowtargetouter{#1}%
  \fi
\fi
}
\newcommand*\glsshowtargetouter}[1]{%
  \glsshowtargetsymbol\marginpar{\glsshowtargetsymbol\glsshowtargetfont #1}}
\newcommand*\glsshowtargetsymbol}{\tiny$\triangleright$}
\newcommand*\glsshowtargetfont}{\ttfamily\footnotesize}
\newcommand*\@glsshowtarget}[1]{%
\newrobustcmd*\@@glsshowtarget}[1]{%
  \begingroup
  \protected@edef\@gls@tmp{#1}%
  \@onelevel@sanitize\@gls@tmp
  \expandafter\glsshowtarget\expandafter{\@gls@tmp}%

```

```

\endgroup
}
\newcommand*{\@glsshowaccsupp}[3]{}
\newrobustcmd*{\glsshowaccsupp}[3]{%
  \ifstrempy{#1}%
  {\@glsshowtarget{/#2 (#3)}}%
  {\@glsshowtarget{/#2 (#3) [#1]}}%
}
\newcommand*{\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries\space (or move
  \string\newglossaryentry\space
  definitions into the preamble)}%
  {You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key. It may
  be that the 'see' key has been written to the .glsdefs
  file from the previous run, in which case you need to
  move your definitions
  to the preamble if you don't want to use
  \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoinde}x{%
  [\gls@seenoinde@val\gls@seenoinde@nr]{error,warn,ignore}{%
    \ifcase\gls@seenoinde@nr
      \renewcommand*{\@gls@see@noindex}{%
        \PackageError{glossaries}%
        {'\gls@xr@key' key may only be used after \string\makeglossaries\space
        or \string\makenoidxglossaries}%
        {You must use \string\makeglossaries\space
        or \string\makenoidxglossaries\space before defining
        any entries that have a '\gls@xr@key' key}}%
      }%
    \or
      \renewcommand*{\@gls@see@noindex}{%
        \GlossariesWarning{'\gls@xr@key' key ignored}%
      }%
    \or
      \renewcommand*{\@gls@see@noindex}{}%
    \fi
  }
\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}{
  \newcommand*{\@glossarysec}{section}}%
  {\newcommand*{\@glossarysec}{chapter}}
\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
subsection,subsubsection,paragraph,subparagraph}[section]{%

```

```

\renewcommand*{\@glossarysec}{#1}}
\newcommand*{\@glossarysecstar}{*}
\newcommand*{\@glossaryseclabel}{}
\newcommand*{\glsautoprefix}{}
\define@choicekey{glossaries.sty}{numberedsection}%
  [\gls@numberedsection@val\gls@numberedsection@nr]{%
false,nolabel,autolabel,nameref}[nolabel]{%
  \ifcase\gls@numberedsection@nr\relax
    \renewcommand*{\@glossarysecstar}{*}%
    \renewcommand*{\@glossaryseclabel}{}%
  \or
    \renewcommand*{\@glossarysecstar}{}%
    \renewcommand*{\@glossaryseclabel}{}%
  \or
    \renewcommand*{\@glossarysecstar}{}%
    \renewcommand*{\@glossaryseclabel}{}%
    \label{\glsautoprefix\@glo@type}}%
  \or
    \renewcommand*{\@glossarysecstar}{*}%
    \renewcommand*{\@glossaryseclabel}{}%
    \protected@edef\@currentlabelname{\glossarytoctitle}%
    \label{\glsautoprefix\@glo@type}}%
  \fi
}
\@ifpackageloaded{classicthesis}
{\newcommand*{\@glossary@default@style}{index}}
{\newcommand*{\@glossary@default@style}{list}}
\define@key{glossaries.sty}{style}{%
  \def\@glossary@default@style{#1}%
}
\newcommand*{\@gls@declareoption}[2]{%
  \DeclareOptionX{#1}{#2}%
  \DeclareOption{#1}{#2}%
}
\newcommand*{\glossaryentrynumbers}[1]{#1\gls@save@numberlist{#1}}
\@gls@declareoption{nonumberlist}{%
  \renewcommand*{\glossaryentrynumbers}[1]{\gls@save@numberlist{#1}}%
}
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{}
\gls@savenumberlistfalse
\newcommand*\@glo@seeautonumberlist{}
\@gls@declareoption{seeautonumberlist}{%
  \renewcommand*\@glo@seeautonumberlist{}%
  \def\@glo@prefix{\glsnextpages}%
}%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{}
\gls@esclocationstrue
\newcommand*{\@gls@loadlong}{\RequirePackage{glossary-long}[=v4.49]}
\@gls@declareoption{no long}{\renewcommand*{\@gls@loadlong}{}

```

```

\IfFileExists{supertabular.sty}{%
  \newcommand*\@gls@loadsuper{\RequirePackage{glossary-super}[=v4.49]}{%
  \newcommand*\@gls@loadsuper{}}
\@gls@declareoption{nosuper}{\renewcommand*\@gls@loadsuper{}}
\newcommand*\@gls@loadlist{\RequirePackage{glossary-list}[=v4.49]}
\@gls@declareoption{no list}{%
  \renewcommand*\@gls@loadlist{%
  \ifdefstring{\@glossary@default@style}{list}%
  {\let\@glossary@default@style\relax}%
  }%
}%
}
\newcommand*\@gls@loadtree{\RequirePackage{glossary-tree}[=v4.49]}
\@gls@declareoption{notree}{\renewcommand*\@gls@loadtree{}}
\@gls@declareoption{nostyles}{%
  \renewcommand*\@gls@loadlong{}%
  \renewcommand*\@gls@loadsuper{}%
  \renewcommand*\@gls@loadlist{}%
  \renewcommand*\@gls@loadtree{}%
  \let\@glossary@default@style\relax
}
\newcommand*\glspostdescription{%
  \ifglsnopostdot\else.\spacefactor\sfcode‘\.\ \fi
}
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{}
\glsnopostdotfalse
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{}
\glsnogroupskipfalse
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{}
\@ifclassloaded{memoir}
{%
  \glsucmarktrue
}%
{%
  \glsucmarkfalse
}
\newcommand*\@gls@define@glossaryentrycounter{%
  \ifglsentrycounter
  \ifundef\c@glossaryentry
  {%
    \ifx\@gls@counterwithin\@empty
    \newcounter{glossaryentry}%
    \else
    \newcounter{glossaryentry}[\@gls@counterwithin]%
    \fi
  \def\theHglossaryentry{\currentglossary.\theglossaryentry}%
  }%
  }%
\fi
}

```

```

\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\glentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
  \renewcommand*{\@gls@counterwithin}{#1}%
  \glentrycountertrue
  \@gls@define@glossaryentrycounter
}
\newcommand*{\@gls@counterwithin}{}
\newcommand{\@gls@define@glossarysubentrycounter}{%
  \ifundef\c@glossarysubentry
  {%
    \ifglssubentrycounter
    \ifglentrycounter
      \newcounter{glossarysubentry}[glossaryentry]%
    \else
      \newcounter{glossarysubentry}%
    \fi
    \def\theH@glossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
  \fi
  }%
  }%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
\glssubentrycounterfalse
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none}{%
  \renewcommand*{\@glo@default@sorttype}{#1}%
  \csname @gls@setupsort@#1\endcsname
}
\newcommand*{\@gls@prestandardsort}[3]{%
  \glsdosanitizesort
}
\newcommand*{\@glo@check@sortallowed}[1]{}
\newcommand*{\@gls@setupsort@standard}{%
  \def\do@glo@storeentry{\@glo@storeentry}%
  \def\@gls@defsortcount##1{%
  \def\@gls@defsort##1##2{%
    \ifx\@glo@sort\@gls@defaultsort
      \let\@glo@sort\@glo@name
    \fi
    \let\glsdosanitizesort\@gls@sanitizesort
    \glsprestandardsort{\@glo@sort}{##1}{##2}%
    \expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
  }%
  \def\@gls@setsort##1{%
  \let\@glo@check@sortallowed\@gobble
}
\@gls@setupsort@standard
\newcommand*\glsortnumberfmt[1]{%
  \ifnum#1<100000 0\fi

```



```

\ifnum#1<10000 0\fi
\ifnum#1<1000 0\fi
\ifnum#1<100 0\fi
\ifnum#1<10 0\fi
\number#1%
}
\newcommand*{\@gls@setupsort@def}{%
\def\do@glo@storeentry{\@glo@storeentry}%
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\ifcsundef{glossary@##1@sortcount}%
{\@gls@defsortcount{##1}}%
{}%
\expandafter\global\expandafter
\advance\csname glossary@##1@sortcount\endcsname by 1\relax
\expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
\expandafter\glssortnumberfmt
{\csname glossary@##1@sortcount\endcsname}}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*{\@gls@setupsort@use}{%
\let\do@glo@storeentry\@gobble
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\expandafter\gdef\csname glo@##2@sort\endcsname{}%
}%
\def\@gls@setsort##1{%
\protected@edef\@glo@parent{\csname glo@##1@parent\endcsname}%
\ifx\@glo@parent\@empty
\else
\expandafter\@gls@setsort\expandafter{\@glo@parent}%
\fi
\protected@edef\@glo@type{\csname glo@##1@type\endcsname}%
\protected@edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
\ifx\@gls@tmp\@empty
\expandafter\global\expandafter
\advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
\expandafter\protected@xdef\csname glo@##1@sort\endcsname{%
\expandafter\glssortnumberfmt
{\csname glossary@\@glo@type @sortcount\endcsname}}%
\@glo@storeentry{##1}%
\fi

```

```

}%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*\@gls@setupsort@none{%
\def\do@glo@storeentry##1{%
\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
\expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort
}%
\def\@gls@setsort##1{%
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
{Option sort=none not allowed with \string##1}%
{(Use sort=def instead)}}%
}
\newcommand*\glsdefmain{%
\if@gls@docloaded
\newglossary[glg2]{main}{gls2}{glo2}{\glossaryname}%
\else
\newglossary{main}{gls}{glo}{\glossaryname}%
\fi
\newcommand*\gls@tr@set@main@toctitle{%
\translatelet{\glossarytoctitle}{Glossary}%
}%
}
\newcommand*\glsdefaulttype{main}
\newcommand*\acronymtype{\glsdefaulttype}
\@gls@declareoption{nomain}{%
\let\glsdefaulttype\relax
\renewcommand*\glsdefmain}{}%
}
\define@boolkey{glossaries.sty}[gls]{acronym}[true]{%
\ifglsacronym
\renewcommand*\@gls@do@acronymsdef{%
\DeclareAcronymList{acronym}%
\newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
\renewcommand*\acronymtype{acronym}%
\newcommand*\gls@tr@set@acronym@toctitle{%
\translatelet{\glossarytoctitle}{Acronyms}%
}%
}%
\else
\let\@gls@do@acronymsdef\relax
\fi
}
\AtBeginDocument{%
\ifglsacronym
\ifbool{glscompatible-3.07}%
{}%
{%
\providecommand*\printacronyms[1]{}%
}
}

```

```

        \printglossary[type=\acronymtype,#1]}%
    }%
\fi
}
\newcommand*{\@gls@do@acronymsdef}{}
\@gls@declareoption{acronyms}{%
  \glsacronymtrue
  \def\@gls@do@acronymsdef{%
    \DeclareAcronymList{acronym}%
    \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
    \renewcommand*{\acronymtype}{acronym}%
    \newcommand*{\gls@tr@set@acronym@toctitle}{%
      \translatelet{\glossarytoctitle}{Acronyms}%
    }%
  }%
}
\newcommand*{\@gls@acronymlists}{}
\newcommand*{\@addtoacronymlists}[1]{%
  \ifx\@gls@acronymlists\@empty
    \protected@xdef\@gls@acronymlists{#1}%
  \else
    \protected@xdef\@gls@acronymlists{\@gls@acronymlists,#1}%
  \fi
}
\newcommand*{\DeclareAcronymList}[1]{%
  \glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}
\newcommand{\glsIfListOfAcronyms}[1]{%
  \protected@edef\@do@gls@islistofacronyms{%
    \noexpand\@gls@islistofacronyms{#1}{\@gls@acronymlists}}%
  \@do@gls@islistofacronyms
}
\newcommand{\@gls@islistofacronyms}[4]{%
  \def\gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%
    \def\@gls@before{##1}\def\@gls@after{##2}}%
  \gls@islistofacronyms,#2,#1,\@nil\end@gls@islistofacronyms
  \ifx\@gls@after\@nnil
    #4%
  \else
    #3%
  \fi
}
\newif\if@gls@isacronymlist
\newcommand*{\gls@checkisacronymlist}[1]{%
  \glsIfListOfAcronyms{#1}%
  {\@gls@isacronymlisttrue}{\@gls@isacronymlistfalse}%
}
\newcommand*{\SetAcronymLists}[1]{%
  \renewcommand*{\@gls@acronymlists}{#1}%
}

```

```

\define@key{glossaries.sty}{acronymlists}{%
  \DeclareAcronymList{#1}%
}
\newcommand{\glscounter}{page}
\define@key{glossaries.sty}{counter}{%
  \renewcommand*{\glscounter}{#1}%
}
\newcommand*{\@gls@nohyperlist}{}
\newcommand*{\GlsDeclareNoHyperList}[1]{%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*{\@gls@nohyperlist}{#1}%
  }%
  {%
    \appto\@gls@nohyperlist{,#1}%
  }%
}
\define@key{glossaries.sty}{nohypertypes}{%
  \GlsDeclareNoHyperList{#1}%
}
\newcommand*{\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{#1}%
}
\newcommand*{\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{#1}%
}
\newcommand{\glosortentrieswarning}{%
  \typeout{Using TeX to sort glossary entries---this may
  take a while}%
}
\@gls@declareoption{nowarn}{%
  \if@gls@debug
    \GlossariesWarning{Warnings can't be suppressed in debug mode}%
  \else
    \renewcommand*{\GlossariesWarning}[1]{}%
    \renewcommand*{\GlossariesWarningNoLine}[1]{}%
    \renewcommand*{\glosortentrieswarning}{}%
    \renewcommand*{\@gls@missinglang@warn}[2]{}%
  \fi
}
\newcommand*{\@gls@missinglang@warn}[2]{%
  \PackageWarningNoLine{glossaries}%
  {No language module detected for '#1'.\MessageBreak
  Language modules need to be installed separately.\MessageBreak
  Please check on CTAN for a bundle called\MessageBreak
  'glossaries-#2' or similar}%
}
\@gls@declareoption{nolangwarn}{%
  \renewcommand*{\@gls@missinglang@warn}[2]{}%
}

```

```

\newcommand*{\@gls@warnonglossdefined}{%
  \GlossariesWarning{Overriding \string\printglossary}%
}
\newcommand*{\@gls@warnontheGLOSSdefined}{%
  \GlossariesWarning{Overriding 'theGLOSS' environment}%
}
\@gls@declareoption{noredefwarn}{%
  \renewcommand*{\@gls@warnonglossdefined}{}%
  \renewcommand*{\@gls@warnontheGLOSSdefined}{}%
}
\newcommand*{\@gls@sanitizedesc}{%
}
\newcommand*{\glssetexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@expand@field{##1}{#1}{##2}%
  }%
}
\newcommand*{\glssetnoexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@noexpand@field{##1}{#1}{##2}%
  }%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}
\glssetnoexpandfield{descplural}
\newcommand*{\@gls@sanitizename}{}
\glssetnoexpandfield{name}
\newcommand*{\@gls@sanitizesymbol}{}
\glssetnoexpandfield{symbol}
\glssetnoexpandfield{symbolplural}
\newcommand*{\@gls@sanitizesort}{%
  \ifgls@sanitizesort
    \@gls@sanitizesort
  \else
    \@gls@nosanitizesort
  \fi
}
\newcommand*{\@gls@sanitizesort}{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@gls@nosanitizesort}{}
\newcommand*{\@gls@noidx@sanitizesort}{%
  \ifdefvoid\@glo@sort
  {}%
  {%
    \expandafter\@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
  \def\@glo@sort{#1#2}%
}

```

```

\@onelevel@sanitize\@glo@sort
}
\newcommand*{\@@gls@noidx@nosanitizesort}{%
\ifdefvoid\@glo@sort
{}%
{}%
\expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
}%
}
\def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
\bgroup
\glsnoidxstripaccents
\protected@xdef\@@glo@sort{#1#2}%
\egroup
\let\@glo@sort\@@glo@sort
}
\newcommand*\glsnoidxstripaccents{%
\let\IeC\@firstofone
\let\add@accent@\@secondoftwo
\let\@text@composite@x\@secondoftwo
\let\@tabacckludge\@secondoftwo
\expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
\let'\@firstofone
\let'\@firstofone
\let~\@firstofone
\let"\@firstofone
\let\u\@firstofone
\let\t\@firstofone
\let\d\@firstofone
\let\r\@firstofone
\let=\@firstofone
\let.\@firstofone
\let~\@firstofone
\letv\@firstofone
\letH\@firstofone
\letc\@firstofone
\letb\@firstofone
\leta\@secondoftwo
\def\AE{AE}%
\def\ae{ae}%
\def\OE{OE}%
\def\oe{oe}%
\def\AA{AA}%
\def\aa{aa}%
\def\L{L}%
\def\l{l}%
\def\O{O}%

```

```

\def\o{o}%
\def\SS{SS}%
\def\ss{ss}%
\def\th{th}%
\def\TH{TH}%
\def\dh{dh}%
\def\DH{DH}%
}
\@ifl@t@r\fmtversion{2019/10/01}
{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}
\define@boolkey[glS]{sanitize}{description}[true]{%
\GlossariesWarning{sanitize={description} package option deprecated}%
\ifglS@sanitize@description
\glssetnoexpandfield{desc}%
\glssetnoexpandfield{descplural}%
\else
\glssetexpandfield{desc}%
\glssetexpandfield{descplural}%
\fi
}
\define@boolkey[glS]{sanitize}{name}[true]{%
\GlossariesWarning{sanitize={name} package option deprecated}%
\ifglS@sanitize@name
\glssetnoexpandfield{name}%
\else
\glssetexpandfield{name}%
\fi
}
\define@boolkey[glS]{sanitize}{symbol}[true]{%
\GlossariesWarning{sanitize={symbol} package option deprecated}%
\ifglS@sanitize@symbol
\glssetnoexpandfield{symbol}%
\glssetnoexpandfield{symbolplural}%
\else
\glssetexpandfield{symbol}%
\glssetexpandfield{symbolplural}%
\fi
}
\define@boolkey{glossaries.sty}[glS]{sanitizesort}[true]{%
\ifglssanitizesort
\glssetnoexpandfield{sortvalue}%
\renewcommand*{\@glS@noidx@setsanitizesort}{%
\glssanitizesorttrue
\glssetnoexpandfield{sortvalue}%
}%
\else
\glssetexpandfield{sortvalue}%

```

```

        \renewcommand*\@gls@noidx@setsanitizesort}{%
            \glssanitizesortfalse
            \glssetexpandfield{sortvalue}%
        }%
    \fi
}
\glssanitizesorttrue
\glssetnoexpandfield{sortvalue}%
\newcommand*\@gls@noidx@setsanitizesort}{%
    \glssanitizesortfalse
    \glssetexpandfield{sortvalue}%
}
\define@choicekey[gls]{sanitize}{sort}[true,false][true]{%
    \setbool{glssanitizesort}{#1}%
    \ifglssanitizesort
        \glssetnoexpandfield{sortvalue}%
    \else
        \glssetexpandfield{sortvalue}%
    \fi
    \GlossariesWarning{sanitize={sort} package option
        deprecated. Use sanitizesort instead}%
}
\define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
    \ifthenelse{\equal{#1}{none}}{%
        {%
            \GlossariesWarning{sanitize package option deprecated}%
            \glssetexpandfield{name}%
            \glssetexpandfield{symbol}%
            \glssetexpandfield{symbolplural}%
            \glssetexpandfield{desc}%
            \glssetexpandfield{descplural}%
        }%
    }%
    \setkeys[gls]{sanitize}{#1}%
}%
}
\newif\ifglstranslate
\newcommand*\@gls@usetranslator{%
    \@ifpackageloaded{polyglossia}%
    {%
        \let\glsifusetranslator\@secondoftwo
    }%
    {%
        \@ifpackageloaded{babel}%
        {%
            \IfFileExists{translator.sty}%
            {%
                \RequirePackage{translator}%
                \let\glsifusetranslator\@firstoftwo
            }%
        }%
    }%
}

```



```

        {}%
    }%
    {}%
}
}
\newcommand{\glsifusedtranslatordict}[3]{%
  \glsifusetranslator
  {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
  {#3}%
}
\@gls@declareoption{nottranslate}{%
  \glstranslatefalse
  \let\@gls@usetranslator\relax
  \let\glsifusetranslator\@secondoftwo
}
\define@choicekey{glossaries.sty}{translate}{%
  [\gls@translate@val\gls@translate@nr]%
  {true,false,babel}[true]%
  {%
    \ifcase\gls@translate@nr\relax
      \glstranslatetrue
      \renewcommand*\@gls@usetranslator{%
        \@ifpackageloaded{polyglossia}%
        {%
          \let\glsifusetranslator\@secondoftwo
        }%
        {%
          \@ifpackageloaded{babel}%
          {%
            \IfFileExists{translator.sty}%
            {%
              \RequirePackage{translator}%
              \let\glsifusetranslator\@firstoftwo
            }%
            {}%
          }%
          {}%
        }%
      }%
    \or
      \glstranslatefalse
      \let\@gls@usetranslator\relax
      \let\glsifusetranslator\@secondoftwo
    \or
      \glstranslatetrue
      \let\@gls@usetranslator\relax
      \let\glsifusetranslator\@secondoftwo
    \fi
  }
  \glstranslatefalse

```

```

\let\glsifusetranslator\@secondoftwo
\@ifpackageloaded{translator}%
{%
  \glstranslatetrue
  \let\glsifusetranslator\@firstoftwo
}%
{%
  \for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
  {
    \@ifpackageloaded{\gls@thissty}%
    {%
      \glstranslatetrue
      \@endfortrue
    }%
    {}%
  }
}
\define@boolkey{glossaries.sty}[gls]{indexonlyfirst}[true]{}
\glsindexonlyfirstfalse
\define@boolkey{glossaries.sty}[gls]{hyperfirst}[true]{}
\glshyperfirsttrue
\newcommand*{\@gls@setacrstyle}{}
\define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{}
\ifbool{glsacrdescription}%
{}%
{%
  \renewcommand*{\@gls@sanitizedesc}{}%
}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{description}[true]{}
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{}
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{}
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{dua}[true]{}
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{shortcuts}[true]{}
\newcommand*{\glsorder}{word}
\newcommand*{\@glsorder}[1]{}
\define@choicekey{glossaries.sty}{order}{word,letter}{}

```

```

\def\glsorder{#1}}
\newif\ifglsxindy
\glsxindyfalse
\@gls@declareoption{makeindex}{\glsxindyfalse}
\define@boolkey[gls]{xindy}{glsnumbers}[true]{%
\gls@xindy@glsnumberstrue
\def\@xdy@main@language{\languagename}%
\define@key[gls]{xindy}{language}{\def\@xdy@main@language{#1}}
\ifcsundef{inputencodingname}{%
\def\gls@codepage{}}{%
\def\gls@codepage{\inputencodingname}
}
\define@key[gls]{xindy}{codepage}{\def\gls@codepage{#1}}
\define@key{glossaries.sty}{xindy}[]{%
\glsxindytrue
\setkeys[gls]{xindy}{#1}%
}
\@gls@declareoption{xindygloss}{%
\glsxindytrue
}
\@gls@declareoption{xindynoglsnumbers}{%
\glsxindytrue
\gls@xindy@glsnumbersfalse
}
\providecommand{\@domakeglossaries}[1]{#1}
\@gls@declareoption{disablemakegloss}{%
\ifdefequal\makeglossaries\@no@makeglossaries
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect
(\string\makenoidxglossaries\space has already been used)}%
}%
{%
\ifdefequal\@makeglossary\@gobble
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect
(\string\makeglossaries\space has already been used)}%
}%
{%
\renewcommand{\@domakeglossaries}[1]{%
\PackageInfo{glossaries}{\string\makeglossaries\space and
\string\makenoidxglossaries\space have been disabled}%
}%
}%
}%
}
\@gls@declareoption{restoremakegloss}{%
\ifdefequal\makeglossaries\@no@makeglossaries
{%
\GlossariesWarning{Option ‘restoremakegloss’ has no effect
(\string\makenoidxglossaries\space has already been used)}%
}
}

```

```

}%
{%
  \ifdefequal\@makeglossary\@gobble
  {%
    \GlossariesWarning{Option ‘restoremakegloss’ has no effect
      (\string\makeglossaries\space has already been used)}%
  }%
  {%
    \PackageInfo{glossaries}{\string\makeglossaries\space and
      \string\makenoidxglossaries\space have been restored}%
    \let\@domakeglossaries\@firstofone
  }%
}%
}
\newcommand*\@do@write@glslabels{%
  \AtEndDocument{\@do@write@glslabels}%
  \let\@do@write@glslabels\relax
}
\newcommand*\@@do@write@glslabels{%
  \newwrite\@glsl@labelsfile
  \immediate\openout\@glsl@labelsfile=\jobname.glslabels
  \protected@edef\@glsl@fulllist{\@glo@types}%
  \ifdefempty{\@ignored@glossaries}{}%
  {\protected@eappto\@glsl@fulllist{,\@ignored@glossaries}}%
  \forallglstentries[\@glsl@fulllist]{\@glstentry}%
  {%
    \ifdefempty{\@glstentry}{}%
    {%
      \ifx\do@write@glslabelsname\@gobble
        \def\@glst@thisentryname{}%
      \else
        \letcs{\@glst@thisentryname}{glo@\@glstetoklabel{\@glstentry}@name}%
        \@onelevel@sanitize\@glst@thisentryname
      \fi
      \immediate\write\@glsl@labelsfile
        {\@glstentry\do@write@glslabelsname{\@glst@thisentryname}}%
    }%
  }%
  \immediate\closeout\@glsl@labelsfile
}
\newcommand*\do@write@glslabelsname}[1]{%
  \begingroup
  \catcode'\^^I12
  \gdef\@write@glslabelsname#1{\^^I#1}
  \endgroup
  \@glsl@declareoption{writeglslabels}{\@do@write@glslabels
    \let\do@write@glslabelsname\@gobble}
  \@glsl@declareoption{writeglslabelnames}{\@do@write@glslabels
    \let\do@write@glslabelsname\@write@glslabelsname}
  \newif\ifglslautomake

```

```

\newcommand{\gls@automake@nr}{1}
\define@choicekey{glossaries.sty}{automake}%
  [\gls@automake@val\gls@automake@nr]{true,false,immediate}[true]{%
  \ifnum\gls@automake@nr=1\relax
    \glsautomakefalse
  \else
    \glsautomaketrue
  \fi
  \ifglsautomake
    \renewcommand*\@gls@doautomake}{%
      \PackageError{glossaries}{You must use
      \string\makeglossaries\space with automake=true}
      {%
        Either remove the automake=true setting or
        add \string\makeglossaries\space to your document preamble.%
      }%
    }%
  \else
    \renewcommand*\@gls@doautomake}{}%
  \fi
}
\glsautomakefalse
\newcommand*\@gls@doautomake{}
\AtEndDocument{\@gls@doautomake}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
  \ifgls savewrites
    \renewcommand*\@gls@writefiles{\@gls@writefiles}%
  \else
    \let\@gls@writefiles\@empty
  \fi
}
\gls savewritesfalse
\let\@gls@writefiles\@empty
\define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{%
\boolfalse{glscompatible-3.07}
\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
  \ifbool{glscompatible-2.07}%
  {%
    \booltrue{glscompatible-3.07}%
  }%
}%
}
\boolfalse{glscompatible-2.07}
\let\@gls@original@makeglossary\makeglossary
\let\@gls@original@glossary@glossary
\def\makeglossary{%
  \GlossariesWarning{Use of \string\makeglossary\space with
  glossaries.sty is \MessageBreak deprecated. Use \string\makeglossaries\space
  instead. If you \MessageBreak need the original definition of
  \string\makeglossary\space use \MessageBreak the package options

```

```

kernelglossredefs=false (to \MessageBreak restore the former definition of
\string\makeglossary) and \MessageBreak nomain (if the file extensions cause a
conflict))%
\makeglossaries
}
\newcommand*{\@gls@override@glossary}[1][main]{%
\GlossariesWarning{Use of \string\glossary\space with
glossaries.sty is deprecated. \MessageBreak Indexing should be performed
with the user level \MessageBreak commands, such as \string\gls\space or
\string\glsadd. If you need the \MessageBreak original definition of
\string\glossary\space use the package \MessageBreak options
kernelglossredefs=false (to restore the \MessageBreak former definition of
\string\glossary) and nomain (if the \MessageBreak file extensions cause a
conflict))%
\gls@glossary{#1}%
}
\if@gls@docloaded
\else
\def\glossary{\@gls@override@glossary}
\fi
\define@choicekey{glossaries.sty}{kernelglossredefs}%
[\@gls@debug@val\@gls@debug@nr]{true,false,nowarn}[true]%
{%
\ifcase\@gls@debug@nr\relax
\def\glossary{\@gls@override@glossary}%
\def\makeglossary{%
\GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is deprecated. Use \string\makeglossaries\space
instead. If you need the original definition of
\string\makeglossary\space use the package options
kernelglossredefs=false (to prevent redefinition of
\string\makeglossary) and nomain (if the file extensions cause a
conflict))%
\makeglossaries
}%
\or
\let\glossary\gls@original@glossary
\let\makeglossary\gls@original@makeglossary
\or
\def\makeglossary{\makeglossaries}%
\renewcommand*{\@gls@override@glossary}[1][main]{%
\gls@glossary{##1}%
}%
\fi
}
\@gls@declareoption{symbols}{%
\let\@gls@do@symbolsdef\@gls@symbolsdef
}
\newcommand*{\@gls@do@symbolsdef}{}
\newcommand*{\@gls@symbolsdef}{%

```

```

\newglossary[slg]{symbols}{sls}{slo}{\glssymbolsgroupname}%
\newcommand*\{printsymbols}[1][\]{\printglossary[type=symbols,##1]}%
\newcommand*\{gls@tr@set@symbols@toctitle}{%
  \translatelet{\glossarytoctitle}{Symbols (glossaries)}%
}%
}%
\@gls@declareoption{numbers}{%
  \let\@gls@do@numbersdef\@gls@numbersdef
}
\newcommand*\{gls@do@numbersdef}{%
\newcommand*\{gls@numbersdef}{%
  \newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
  \newcommand*\{printnumbers}[1][\]{\printglossary[type=numbers,##1]}%
  \newcommand*\{gls@tr@set@numbers@toctitle}{%
    \translatelet{\glossarytoctitle}{Numbers (glossaries)}%
  }%
}%
\@gls@declareoption{index}{%
  \ifx\@gls@do@indexdef\@empty
    \let\@gls@do@indexdef\@gls@indexdef
  \fi
}
\@gls@declareoption{noglossaryindex}{%
  \let\@gls@do@indexdef\relax
}
\newcommand*\{gls@do@indexdef}{%
\newcommand*\{gls@indexdef}{%
  \newglossary[ilg]{index}{ind}{idx}{\indexname}%
  \newcommand*\{printindex}[1][\]{\printglossary[type=index,##1]}%
  \newcommand*\{newterm}[2][\]{%
    \newglossaryentry{##2}%
    {type={index},name={##2},description={\nopostdesc},##1}}
  \let\@gls@do@indexdef\relax
}%
\@for\CurrentOption :=\@declaredoptions\do{%
  \ifx\CurrentOption\@empty
  \else
    \@expandtwoargs
    \in@ {,\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
    \ifin@
      \@use@ption
      \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
    \fi
  \fi
}
\ProcessOptionsX
\RequirePackage{glossaries-compatible-307}[v4.49]
\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,notranslate,nolong,nosuper,notree,nostyles,%

```

```

nomain,noglossaryindex}
\newcommand*\setupglossaries[1]{%
  \renewcommand*\@gls@setacrstyle}{}%
  \ifglsacrshortcuts
    \def\@gls@setupshortcuts{\glsacrshortcutstrue}%
  \else
    \def\@gls@setupshortcuts{%
      \ifglsacrshortcuts
        \DefineAcronymSynonyms
      \fi
    }%
  \fi
  \glsacrshortcutsfalse
  \let\@gls@do@numbersdef\relax
  \let\@gls@do@symbolssdef\relax
  \let\@gls@do@indexdef\relax
  \let\@gls@do@acronymsdef\relax
  \ifglsentrycounter
    \let\@gls@doentrycounterdef\relax
  \else
    \let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
  \fi
  \ifglssubentrycounter
    \let\@gls@dosubentrycounterdef\relax
  \else
    \let\@gls@dosubentrycounterdef\@gls@define@glossarysubentrycounter
  \fi
  \setkeys{glossaries.sty}{#1}%
  \@gls@setacrstyle
  \@gls@setupshortcuts
  \@gls@do@acronymsdef
  \@gls@do@numbersdef
  \@gls@do@symbolssdef
  \@gls@do@indexdef
  \@gls@doentrycounterdef
  \@gls@dosubentrycounterdef
}
\ifthenelse{\equal{\glscounter}{section}}{%
  {%
    \ifcsundef{chapter}{}%
    {%
      \let\@gls@old@chapter\@chapter
      \def\@chapter[#1]#2{\@gls@old@chapter[#1]}{#2}%
      \ifcsundef{hyperdef}{\hyperdef{section}{\thesection}}{}}%
    }%
  }%
  {}
\newcommand*\@gls@onlypremakeg{}
\newcommand*\@onlypremakeg[1]{%
  \ifx\@gls@onlypremakeg\@empty

```



```

        \def\@gls@onlypremakeg{#1}%
    \else
        \expandafter\toks@\expandafter{\@gls@onlypremakeg}%
        \edef\@gls@onlypremakeg{\the\toks@,\noexpand#1}%
    \fi
}
\newcommand*{\@disable@onlypremakeg}{%
\@for\@thiscs:=\@gls@onlypremakeg\do{%
    \expandafter\@disable@premakecs\@thiscs%
}}
\newcommand*{\@disable@premakecs}[1]{%
    \def#1{\PackageError{glossaries}{\string#1\space may only be
    used before \string\makeglossaries}{You can't use
    \string#1\space after \string\makeglossaries}}%
}
\providecommand*\glossaryname{Glossary}
\providecommand*\acronymname{Acronyms}
\newcommand*\glssettoctitle[1]{%
    \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}
\providecommand*\entryname{Notation}
\providecommand*\descriptionname{Description}
\providecommand*\symbolname{Symbol}
\providecommand*\pagelistname{Page List}
\providecommand*\glsymbolsgroupname{Symbols}
\providecommand*\glsnumbersgroupname{Numbers}
\newcommand*\glspluralsuffix{s}
\newcommand*\glsacrpluralsuffix{\glspluralsuffix}
\newcommand*\glsupacrpluralsuffix{\glsup{glsacrpluralsuffix}}
\providecommand*\seename{see}
\providecommand*\andname{\&}
\newcommand*\RequireGlossariesLang[1]{%
    \ifundefined{ver@glossaries-#1.ldf}{\input{glossaries-#1.ldf}}{}}
}
\newcommand*\ProvidesGlossariesLang[1]{%
    \ProvidesFile{glossaries-#1.ldf}%
}
\newcommand*\addglossarytocaptions[1]{}
\ifglstranslate
    \RequirePackage{tracklang}
    \@gls@usetranslator
    \ifpackageloaded{translator}
    {%
        \ifboolexpr
        {
            test {\ifdefstring{\trans@languages}{English}}
            and not
            test {\ifdefstring{bbl@loaded}{english}}
        }
        {%
            \let\glsifusetranslator\@secondoftwo

```

```

}%
{%
  \usedictionary{glossaries-dictionary}%
  \renewcommand*{\addglossarytocaptions}[1]{%
    \ifcsundef{captions#1}{}%
    {%
      \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
      \expandafter\toks@\expandafter{\@gls@tmp
        \renewcommand*{\glossaryname}{\translate{Glossary}}}%
    }%
    \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
  }%
}%
}%
{}%
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
  }%
}%
{}%
\glsifusetranslator
{%
  \renewcommand*{\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*{\glossaryname}{\translate{Glossary}}%
  \renewcommand*{\acronymname}{\translate{Acronyms}}%
  \renewcommand*{\entryname}{\translate{Notation (glossaries)}}%
  \renewcommand*{\descriptionname}{%
    \translate{Description (glossaries)}}%
  \renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}%
  \renewcommand*{\pagelistname}{%
    \translate{Page List (glossaries)}}%
}

```

```

        \renewcommand*\glssymbolsgroupname}{%
          \translate{Symbols (glossaries)}}%
        \renewcommand*\glsnumbersgroupname}{%
          \translate{Numbers (glossaries)}}%
      }{}%
\fi
\DeclareRobustCommand*\nopostdesc{}
\newcommand*\@nopostdesc}{%
  \let\org@glspostdescription\glspostdescription
  \def\glspostdescription{%
    \let\glspostdescription\org@glspostdescription}%
}
\newcommand*\@no@post@desc}{\nopostdesc}
\newcommand*\glspar}{\par}
\newcommand*\setStyleFile[1]{%
  \renewcommand*\gls@istfilebase}{#1}%
  \ifglxindy
    \def\istfilename{\gls@istfilebase.xdy}
  \else
    \def\istfilename{\gls@istfilebase.ist}
  \fi
}
\@onlypremakeg\setStyleFile
\ifglxindy
  \def\istfilename{\gls@istfilebase.xdy}
\else
  \def\istfilename{\gls@istfilebase.ist}
\fi
\newcommand*\gls@istfilebase}{\jobname}
\newcommand*\@istfilename}[1]{}
\newcommand*\glscompositor}{.}
\newcommand*\glsSetCompositor}[1]{%
  \renewcommand*\glscompositor}{#1}}
\@onlypremakeg\glsSetCompositor
\newcommand*\@glsAlphaCompositor}{\glscompositor}
\ifglxindy
  \newcommand*\glsSetAlphaCompositor[1]{%
    \renewcommand*\@glsAlphaCompositor}{#1}}
\else
  \newcommand*\glsSetAlphaCompositor[1]{%
    \glsnoxindywarning\glsSetAlphaCompositor}
\fi
\@onlypremakeg\glsSetAlphaCompositor
\newcommand*\gls@suffixF}{}
\newcommand*\glsSetSuffixF}[1]{%
  \renewcommand*\gls@suffixF}{#1}}
\@onlypremakeg\glsSetSuffixF
\newcommand*\gls@suffixFF}{}
\newcommand*\glsSetSuffixFF}[1]{%
  \renewcommand*\gls@suffixFF}{#1}}

```

```

}
\ifcsundef{hyperlink}%
{%
  \newcommand*\glsnumberformat}[1]{#1}%
}%
{%
  \newcommand*\glsnumberformat}[1]{\glsnumberformat{#1}}%
}
\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*\glossarypreamble{%
  \csuse{@glossarypreamble@\currentglossary}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaulttype]{%
  \ifglossaryexists*{#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*\glossarypostamble{}
\newcommand*\glossarysection}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@p@glossarysection{#1}{#2}%
  }%
  \gls@glossarymark{\glossarytoctitle}%
}
\ifcsundef{glossarymark}%
{%
  \newcommand*\gls@glossarymark}[1]{\glossarymark{#1}}
}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand*\gls@glossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUchead{#1}}{\memUchead{#1}}%
      \else
        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand*\gls@glossarymark}[1]{%
      \ifglsucmark
        \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
      \else

```

```

        \@mkboth{#1}{#1}%
    \fi
}
}
\providecommand{\glossarymark}[1]{%
    \ifglsucmark
        \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
    \else
        \@mkboth{#1}{#1}%
    \fi
}
\newcommand*\setglossarysection[1]{%
\setkeys{glossaries.sty}{section=#1}}
\newcommand*{@glossarysection}[2]{%
    \ifdefempty{@glossarysecstar
    {%
        \csname@@glossarysec\endcsname[#1]{#2}%
    }%
    {%
        \csname@@glossarysec\endcsname*{#2}%
        \@gls@toc{#1}{\@@glossarysec}%
    }%
    \@glossaryseclabel
}
\newcommand*{@p@glossarysection}[2]{%
    \glsclearpage
    \phantomsection
    \ifdefempty{@glossarysecstar
    {%
        \csname@@glossarysec\endcsname{#2}%
    }%
    {%
        \@gls@toc{#1}{\@@glossarysec}%
        \csname@@glossarysec\endcsname*{#2}%
    }%
    \@glossaryseclabel
}
\newcommand*\gls@docclearpage{%
    \ifthenelse{\equal{\@@glossarysec}{chapter}}%
    {%
        \ifcsundef{cleardoublepage}%
        {%
            \clearpage
        }%
        {%
            \ifcsdef{if@openright}%
            {%
                \if@openright
                \cleardoublepage
            }%
        }%
    }%
}

```

```

        \else
        \clearpage
        \fi
    }%
    {%
    \cleardoublepage
    }%
    }%
    }%
    {}%
}
\newcommand*\glsclearpage{\gls@docclearpage}
\newcommand*\@gls@toc}[2]{%
    \ifglstoc
        \ifglsnumberline
            \addcontentsline{toc}{#2}{\protect\numberline{#1}}%
        \else
            \addcontentsline{toc}{#2}{#1}%
        \fi
    \fi
}
\newcommand*\glsnoxindywarning}[1]{%
    \GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
}
\newcommand*\glsnomakeindexwarning}[1]{%
    \GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%
}
\ifglsxindy
    \edef\xdyattributes{\string"default\string"}%
\fi
\ifglsxindy
    \edef\xdyattributelist{}%
\fi
\ifglsxindy
    \def\xdylocref{}
\fi
\newcommand*\@gls@ifinlist}[4]{%
    \def\@do@ifinlist##1,#1,##2\end@ifinlist{%
        \def\@gls@listsuffix{##2}%
        \ifx\@gls@listsuffix\@empty
            #4%
        \else
            #3%
        \fi
    }%
    \@do@ifinlist,#2,#1,\end@ifinlist
}
\ifglsxindy
    \newcommand*\@xdycounters{\glscounter}
    \newcommand*\GlsAddXdyCounters[1]{%

```

```

\@for\@gls@ctr:=#1\do{%
  \edef\@do@addcounter{%
    \noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
    {%
      \noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
        \noexpand\@gls@ctr}%
    }%
  }%
  \@do@addcounter
}
}
\@onlypremakeg\GlsAddXdyCounters
\else
\newcommand*\GlsAddXdyCounters[1]{%
  \glsnoxindywarning\GlsAddXdyAttribute
}
\fi
\newcommand*\@disabled@glsaddxdycounters{%
  \PackageError{glossaries}{\string\GlsAddXdyCounters\space
    can't be used after \string\GlsAddXdyAttribute}{Move all
    occurrences of \string\GlsAddXdyCounters\space before the first
    instance of \string\GlsAddXdyAttribute}%
}
\ifglsxindy
\newcommand*\@glsaddxdyattribute[2]{%
  \edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string" ^^J
    \string"#2#1\string"}%
  \expandafter\toks@\expandafter{\@xdylocref}%
  \edef\@xdylocref{\the\toks@ ^^J}%
  (markup-locref
  :open \string"\glstildechar n%
    \expandafter\string\csname glsX#2X#1\endcsname
    \string" ^^J
  :close \string"\string" ^^J
  :attr \string"#2#1\string"))%
  \expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
    \setentrycounter{##1}{#2}\csname #1\endcsname{##2}%
  }%
}
\newcommand*\GlsAddXdyAttribute[1]{%
  \ifx\@xdyattributelist\@empty
    \edef\@xdyattributelist{#1}%
  \else
    \edef\@xdyattributelist{\@xdyattributelist,#1}%
  \fi
  \@for\@this@counter:=\@xdycounters\do{%
    \protected@edef\gls@do@addxdyattribute{%
      \noexpand\@glsaddxdyattribute{#1}{\@this@counter}%
    }
    \gls@do@addxdyattribute
  }
}

```

```

    }%
    \let\GlsAddXdyCounters\@disabled@gl saddx dycounters
  }
  \@onlypremakeg\GlsAddXdyAttribute
\else
  \newcommand*\GlsAddXdyAttribute[1]{%
    \gl snoxindywarning\GlsAddXdyAttribute}
\fi
\ifgl sxindy
\newcommand*{\@gl s@addpredefinedattributes}{%
  \GlsAddXdyAttribute{gl snumberformat}
  \GlsAddXdyAttribute{textrm}
  \GlsAddXdyAttribute{textsf}
  \GlsAddXdyAttribute{texttt}
  \GlsAddXdyAttribute{textbf}
  \GlsAddXdyAttribute{textmd}
  \GlsAddXdyAttribute{textit}
  \GlsAddXdyAttribute{textup}
  \GlsAddXdyAttribute{textsl}
  \GlsAddXdyAttribute{textsc}
  \GlsAddXdyAttribute{emph}
  \GlsAddXdyAttribute{gl shypernumber}
  \GlsAddXdyAttribute{hyperm}
  \GlsAddXdyAttribute{hypersf}
  \GlsAddXdyAttribute{hyper tt}
  \GlsAddXdyAttribute{hyperbf}
  \GlsAddXdyAttribute{hypermd}
  \GlsAddXdyAttribute{hyperit}
  \GlsAddXdyAttribute{hyperup}
  \GlsAddXdyAttribute{hypersl}
  \GlsAddXdyAttribute{hypersc}
  \GlsAddXdyAttribute{hyperemph}
  \GlsAddXdyAttribute{gl signore}
}
\else
  \let\@gl s@addpredefinedattributes\relax
\fi
\def\@xdyuseralphabets{}
\ifgl sxindy
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \edef\@xdyuseralphabets{%
      \@xdyuseralphabets ^~J
      (define-alphabet "#1" (#2))}}
\else
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \gl snoxindywarning\GlsAddXdyAlphabet}
\fi
\ifgl sxindy
  \newcommand*{\@gl s@xdy@locationlist}{%
    roman-page-numbers,%

```



```

    Roman-page-numbers,%
    arabic-page-numbers,%
    alpha-page-numbers,%
    Alpha-page-numbers,%
    Appendix-page-numbers,%
    arabic-section-numbers%
}
\protected@edef\@gls@roman{\@roman{0}\string"
  \string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@gls@roman
\edef\@gls@tmp{\string" \string"roman-numbers-lowercase\string"
  :sep \string"}%
\@onelevel@sanitize\@gls@tmp
\ifx\@gls@tmp\@gls@roman
  \expandafter
    \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
      \string"roman-numbers-lowercase\string"%
    }%
\else
  \expandafter
    \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
      :sep \string"\@gls@roman\string"%
    }%
\fi
\expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
  \string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
  \string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
  \string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
  \string"ALPHA\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
  \string"ALPHA\string"
  :sep \string"\@gls@alphacompositor\string"
  \string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
  \string"arabic-numbers\string"
  :sep \string"\glscompositor\string"
  \string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglxindy

```

```

\newcommand\@xdycrossrefhook{}
\fi
\ifglxindy
\newcommand*\GlsAddXdyLocation}[3][{}]{%
\def\@gls@tmp{#1}%
\ifx\@gls@tmp\@empty
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep \string"{}\glsopenbrace\string" #3
:sep \string"\glsclosebrace\string"))
}%
\else
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep "\glsopenbrace"
#1
:sep "\glsclosebrace\glsopenbrace" #3
:sep "\glsclosebrace"))
}%
\fi
\edef\@xdyuserlocationnames{%
\@xdyuserlocationnames^^J\space\space\space
\string"#2\string"}%
}
\@onlypremakeg\GlsAddXdyLocation
\else
\newcommand*\GlsAddXdyLocation}[2]{%
\glsnoxywarning\GlsAddXdyLocation}
\fi
\ifglxindy
\def\@xdylocationclassorder{^^J\space\space\space
\string"roman-page-numbers\string"^^J\space\space\space
\string"arabic-page-numbers\string"^^J\space\space\space
\string"arabic-section-numbers\string"^^J\space\space\space
\string"alpha-page-numbers\string"^^J\space\space\space
\string"Roman-page-numbers\string"^^J\space\space\space
\string"Alpha-page-numbers\string"^^J\space\space\space
\string"Appendix-page-numbers\string"
\@xdyuserlocationnames^^J\space\space\space
\string"see\string"}
\fi
\ifglxindy
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\def\@xdylocationclassorder{#1}}
\else
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\glsnoxywarning\GlsSetXdyLocationClassOrder}

```

```

\fi
\ifglxindy
  \def\@xdysortrules{}
\fi
\ifglxindy
  \newcommand*\GlsAddSortRule[2]{%
    \expandafter\toks@\expandafter{\@xdysortrules}%
    \protected@edef\@xdysortrules{\the\toks@ ^^J
      (sort-rule \string"#1\string" \string"#2\string")}%
    }
\else
  \newcommand*\GlsAddSortRule[2]{%
    \glsnoxywarning\GlsAddSortRule}
\fi
\ifglxindy
  \def\@xdyrequiredstyles{tex}
\fi
\ifglxindy
  \newcommand*\GlsAddXdyStyle[1]{%
    \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
  \newcommand*\GlsAddXdyStyle[1]{%
    \glsnoxywarning\GlsAddXdyStyle}
\fi
\ifglxindy
  \newcommand*\GlsSetXdyStyles[1]{%
    \edef\@xdyrequiredstyles{#1}}
\else
  \newcommand*\GlsSetXdyStyles[1]{%
    \glsnoxywarning\GlsSetXdyStyles}
\fi
\newcommand*\findrootlanguage{-}
\def\@xdylanguage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
    \ifglossaryexists{#1}{%
      \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
    }{%
      \PackageError{glossaries}{Can't set language type for
        glossary type '#1' --- no such glossary}{%
        You have specified a glossary type that doesn't exist}}
\else
  \newcommand*\GlsSetXdyLanguage[2][ ]{%
    \glsnoxywarning\GlsSetXdyLanguage}
\fi
\def\@gls@codepage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyCodePage[1]{%
    \renewcommand*\@gls@codepage{#1}%
  }

```

```

\AtBeginDocument{%
  \ifx\gls@codepage\empty
    \ifpackageloaded{fontspec}{\def\gls@codepage{utf8}}{}%
  \fi
}
\else
  \newcommand*\GlsSetXdyCodePage[1]{%
    \glsnoxindywarning\GlsSetXdyCodePage}
\fi
\ifglsxindy
  \ifgls@xindy@glsnumbers
    \def\@xdylettergroups{(define-letter-group
      \string\glsnumbers\string^^J\space\space\space
      :prefixes (\string"0\string" \string"1\string"
        \string"2\string" \string"3\string" \string"4\string"
        \string"5\string" \string"6\string" \string"7\string"
        \string"8\string" \string"9\string")^^J\space\space\space
      \@xdynumbergrouporder)}
    \else
      \def\@xdylettergroups{}
    \fi
\fi
\newcommand*\GlsAddLetterGroup[2]{%
  \expandafter\toks@\expandafter{\@xdylettergroups}%
  \protected@edef\@xdylettergroups{\the\toks@^^J%
    (define-letter-group \string"#1\string"^^J\space\space\space#2)}%
  }%
\newcommand*\forallglossaries[3][\@glo@types]{%
  \@for#2:=#1\do{\ifx#2\empty\else#3\fi}%
}

\newcommand*\forallacronyms[2]{%
  \@for#1:=\@glsacronymlists\do{\ifx#1\empty\else#2\fi}%
}
\newcommand*\forglentries[3][\glsdefaulttype]{%
  \protected@edef\@glo@list{\csname glolist@#1\endcsname}%
  \@for#2:=\@glo@list\do
  {%
    \ifdefempty{#2}{#3}%
  }%
}
\newcommand*\forallglentries[3][\@glo@types]{%
  \expandafter\forallglossaries\expandafter[#1]{\@@this@glo@}%
  {%
    \forglentries[\@@this@glo@]{#2}{#3}%
  }%
}
\newcommand\ifglossaryexists{%
  \@ifstar\s@ifglossaryexists\@ifglossaryexists
}

```

```

\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{@glotype@#1@out}{#3}{#2}%
}
\newcommand{\s@ifglossaryexists}[3]{%
  \ifcsundef{glolist@#1}{#3}{#2}%
}
\newcommand*{\glsdetoklabel}[1]{#1}
\newcommand{\ifglsentryexists}[3]{%
  \ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}
\newcommand*{\ifglsused}[3]{%
  \ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}
\newcommand{\glsdoifexists}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
}
\newcommand{\glsdoifnoexists}[2]{%
  \ifglsentryexists{#1}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
      been defined}{}}{#2}%
}
\newcommand{\glsdoifexistsorwarn}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}%
  }%
}
\newcommand{\glsdoifexistsordo}[3]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
  #3%
}%
}
\newcommand{\doifglossarynoexistsordo}[3]{%
  \ifglossaryexists*{#1}%
  {%
    \PackageError{glossaries}{Glossary type ‘#1’ already exists}{}%
    #3%
  }%
  {#2}%
}
\newrobustcmd{\ifglshaschildren}[3]{%
  \glsdoifexists{#1}%
  {%
    \def\do@glshaschildren{#3}%
  }
}

```

```

\protected@edef\@gls@thislabel{\glsdetoklabel{#1}}%
\expandafter\forglentries\expandafter
  [\csname glo@\@gls@thislabel @type\endcsname]
{\glo@label}%
{%
  \letcs\glo@parent{glo@\glo@label @parent}%
  \ifdefequal\@gls@thislabel\glo@parent
  {%
    \def\do@glshaschildren{#2}%
    \@endfortrue
  }%
  }%
  \do@glshaschildren
}%
}
\newcommand{\ifglshasparent}[3]{%
  \glsdoifexists{#1}%
  {%
    \ifcseempty{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
  }%
}
\newcommand*\{ifglshasdesc}[3]{%
  \ifcseempty{glo@\glsdetoklabel{#1}@desc}%
  {#3}%
  {#2}%
}
\newcommand*\{ifglshasdescsuppressed}[3]{%
  \ifcsequal{glo@\glsdetoklabel{#1}@desc}{\no@post@desc}%
  {#2}%
  {#3}%
}
\newrobustcmd*\{ifglshassymbol}[3]{%
  \letcs{\@glo@symbol}{glo@\glsdetoklabel{#1}@symbol}%
  \ifdefempty\@glo@symbol
  {#3}%
  {%
    \ifdefequal\@glo@symbol\@gls@default@value
    {#3}%
    {#2}%
  }%
}
\newrobustcmd*\{ifglshaslong}[3]{%
  \letcs{\@glo@long}{glo@\glsdetoklabel{#1}@long}%
  \ifdefempty\@glo@long
  {#3}%
  {%
    \ifdefequal\@glo@long\@gls@default@value
    {#3}%
    {#2}%
  }%
}

```

```

    }%
  }
\newrobustcmd*{\ifglshasshort}[3]{%
  \letcs{\@glo@short}{glo@glsetoklabel{#1}@short}%
  \ifdefempty\@glo@short
    {#3}%
  {%
    \ifdefequal\@glo@short\@gls@default@value
      {#3}%
    {#2}%
  }%
}
}
\newrobustcmd*{\ifglshasfield}[4]{%
  \glsdoifexists{#2}%
  {%
    \letcs{\@glo@thisvalue}{glo@glsetoklabel{#2}@#1}%
    \ifdef\@glo@thisvalue
      {%
        \ifdefempty\@glo@thisvalue
          {%
            #4%
          }%
        {%
          \ifdefequal\@glo@thisvalue\@gls@default@value
            {%
              #4%
            }%
          {%
            \let\glscurrentfieldvalue\@glo@thisvalue
            #3%
          }%
        }%
      }%
    }%
    {%
      \@gls@fetchfield{\@gls@thisfield}{#1}%
      \ifdef\@gls@thisfield
        {%
          \letcs{\@glo@thisvalue}{glo@glsetoklabel{#2}@#1}%
          \ifdefempty\@glo@thisvalue
            {%
              #4%
            }%
          {%
            \ifdefequal\@glo@thisvalue\@gls@default@value
              {%
                #4%
              }%
            {%
              \let\glscurrentfieldvalue\@glo@thisvalue
              #3%
            }%
          }%
        }%
      }%
    }%
  }%
}

```



```

\expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
\expandafter\@gls@forbidtextext\csname @glotype@#2@log\endcsname
\expandafter\@gls@forbidtextext\csname @glotype@#2@in\endcsname
\expandafter\@gls@forbidtextext\csname @glotype@#2@out\endcsname
\expandafter\def\csname @glotype@#2@title\endcsname{#5}%
\@gls@provide@newglossary
\protected@write\@auxout{}\{\string\@newglossary{#2}{#1}{#3}{#4}}%
\ifcsundef{gls@#2@entryfmt}%
{%
  \defglsentryfmt[#2]{\glsentryfmt}%
}%
}%
\@gls@defsortcount{#2}%
\@ifnextchar[{\@gls@setcounter{#2}}%
  {\@gls@setcounter{#2}[\glscounter]}%
}%
{%
  \gls@gobbleopt
}%
}
\newcommand*\altnewglossary}[3]{%
  \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
}
\@onlypreamble{\newglossary}
\@onlypremakeg\newglossary
\newcommand*\@newglossary}[4]{
\def\@gls@setcounter#1[#2]{%
  \expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
  \ifglxindy
    \GlsAddXdyCounters{#2}%
  \fi
}
\newcommand*\@gls@getcounter}[1]{%
  \csname @glotype@#1@counter\endcsname
}
\glsdefmain
\@gls@do@acronymsdef
\@gls@do@symbolsdef
\@gls@do@numbersdef
\@gls@do@indexdef
\newcommand*\@newignoredglossary}[1]{%
  \ifdefempty\@ignored@glossaries
  {%
    \protected@edef\@ignored@glossaries{#1}%
  }%
  {%
    \protected@eappto\@ignored@glossaries{,#1}%
  }%
  \csgdef{glolist@#1}{,}%
  \ifcsundef{gls@#1@entryfmt}%

```

```

{%
  \defglsentryfmt[#1]{\glsentryfmt}%
}%
{}%
\ifdefempty\@gls@nohyperlist
{%
  \renewcommand*\@gls@nohyperlist{#1}%
}%
{%
  \protected@eappto\@gls@nohyperlist{, #1}%
}%
}
\newcommand*\@ignored@glossaries{}
\newcommand*\@ifignoredglossary[3]{%
  \protected@edef\@gls@igtype{#1}%
  \expandafter\DTLifinlist\expandafter
  {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
}
\define@key{glossentry}{name}{%
\def\@glo@name{#1}%
}
\define@key{glossentry}{description}{%
\def\@glo@desc{#1}%
}
\define@key{glossentry}{descriptionplural}{%
\def\@glo@descplural{#1}%
}
\define@key{glossentry}{sort}{%
\def\@glo@sort{#1}}
\define@key{glossentry}{text}{%
\def\@glo@text{#1}%
}
\define@key{glossentry}{plural}{%
\def\@glo@plural{#1}%
}
\define@key{glossentry}{first}{%
\def\@glo@first{#1}%
}
\define@key{glossentry}{firstplural}{%
\def\@glo@firstplural{#1}%
}
\newcommand*\@gls@default@value{\relax}
\define@key{glossentry}{symbol}{%
\def\@glo@symbol{#1}%
}
\define@key{glossentry}{symbolplural}{%
\def\@glo@symbolplural{#1}%
}
\define@key{glossentry}{type}{%
\def\@glo@type{#1}}

```

```

\define@key{glossentry}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
}%
\define@key{glossentry}{see}{%
  \gls@set@xr@key{see}{\@glo@see}{#1}%
}
\newcommand*{\gls@set@xr@key}[3]{%
  \renewcommand*{\gls@xr@key}{#1}%
  \gls@checkseeallowed
  \def#2{#3}%
  \@glo@seeautonumberlist
}
\newcommand*{\gls@xr@key}{see}
\newcommand*{\gls@checkseeallowed}{%
  \@gls@see@noindex
}
\newcommand*{\gls@checkseeallowed@preambleonly}{%
  \GlossariesWarning
  {‘\gls@xr@key’ key doesn’t have any effect when used in the document
  environment. Move the definition to the preamble
  after \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@key{glossentry}{parent}{%
  \def\@glo@parent{#1}}
\define@choicekey{glossentry}{nonumberlist}{%
  [\@gls@nonumberlist@val\@gls@nonumberlist@nr]{true,false}[true]%
}{%
  \ifcase\@gls@nonumberlist@nr\relax
    \def\@glo@prefix{\@gls@nonextpages}%
    \@gls@savenonumberlist{true}%
  \else
    \def\@glo@prefix{\@gls@nextpages}%
    \@gls@savenonumberlist{false}%
  \fi
}
\newcommand*{\@gls@savenonumberlist}[1]{%
\newcommand*{\@gls@initnonumberlist}{%
\newcommand*{\@gls@storenonumberlist}[1]{%

```

```

\newcommand*\@gls@enablesavenonumberlist}{%
  \renewcommand*\@gls@initnonumberlist}{%
    \undef\@glo@nonumberlist
  }%
  \renewcommand*\@gls@savenonumberlist}[1]{%
    \def\@glo@nonumberlist{##1}%
  }%
  \renewcommand*\@gls@storenonumberlist}[1]{%
    \ifdef\@glo@nonumberlist
      {%
        \cslet{glo@\glsdetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
      }%
    }%
  }%
  \appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}
\define@key{glossentry}{user1}{%
  \def\@glo@useri{#1}%
}
\define@key{glossentry}{user2}{%
  \def\@glo@userii{#1}%
}
\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#1}%
}
\define@key{glossentry}{short}{%
  \def\@glo@short{#1}%
}
\define@key{glossentry}{shortplural}{%
  \def\@glo@shortpl{#1}%
}
\define@key{glossentry}{long}{%
  \def\@glo@long{#1}%
}
\define@key{glossentry}{longplural}{%
  \def\@glo@longpl{#1}%
}
\newcommand*\@gls@noname}{%
  \PackageError{glossaries}{name key required in
  \string\newglossaryentry\space for entry '\@glo@label'}{You
  haven't specified the entry name}
}

```

```

\newcommand*\@glsnodesc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '\@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}%
\newcommand*{\@glsdefaultplural}{}
\newcommand*{\@gls@missingnumberlist}[1]{%
  ??%
  \ifglssavenumberlist
  \GlossariesWarning{Missing number list for entry '#1'.
    Maybe makeglossaries + rerun required}%
  \else
  \PackageError{glossaries}%
  {Package option 'savenumberlist=true' required}%
  {%
    You must use the 'savenumberlist' package option
    to reference location lists.%
  }%
  \fi
}
\newcommand*{\@glsdefaultsort}{\@glo@name}
\newcount\gls@level
\newcommand{\@gls@noexpand@field}[3]{%
  \expandafter\global\expandafter
  \let\csname glo@#1@#2\endcsname#3%
}
\newcommand{\@gls@noexpand@fields}[4]{%
  \ifcsdef{gls@assign@#3@field}
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
    }%
    {%
      \csuse{gls@assign@#3@field}{#2}{#4}%
    }%
  }%
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \@gls@noexpand@field{#2}{#3}{\@gls@value}%
    }%
    {%
  }
}

```

```

        \@@gls@noexpand@field{#2}{#3}{#4}%
    }%
}
\newcommand{\@@gls@expand@field}[3]{%
\expandafter
\protected@xdef\csname glo@#1@#2\endcsname{#3}%
}
\newcommand{\@gls@expand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
{%
\@@gls@expand@field{#2}{#3}{#4}%
}%
{%
\csuse{gls@assign@#3@field}{#2}{#4}%
}%
}%
}%
\ifdefequal{#4}{\@gls@default@value}%
{%
\@@gls@expand@field{#2}{#3}{#1}%
}%
{%
\@@gls@expand@field{#2}{#3}{#4}%
}%
}%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
\def\@gls@tmp{#1}%
\ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*\{gls@expandfields}{%
\let\gls@assign@field\@gls@expand@fields
}
\newcommand*\{gls@noexpandfields}{%
\let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
\glsdoifnoexists{#1}%

```

```

    {%
      \gls@defglossaryentry{#1}{#2}%
    }%
  }
\newcommand*\gls@defdocnewglossaryentry{%
  \let\newglossaryentry\new@glossaryentry
}
\newrobustcmd{\provideglossaryentry}[2]{%
  \ifglsentryexists{#1}%
  {%
    {%
      \gls@defglossaryentry{#1}{#2}%
    }%
  }
}
\@onlypreamble{\provideglossaryentry}
\newcommand*\gls@glsdefs@labellist{}
\newcommand*\gls@glsdefs@addlabel}[1]{\listxadd{\gls@glsdefs@labellist}{#1}}
\newcommand*\gls@docdef@labellist{}
\newcommand*\gls@docdef@addlabel}[1]{\listxadd{\gls@docdef@labellist}{#1}}
\newrobustcmd{\new@glossaryentry}[2]{%
  \ifundef\@gls@deffile
  {%
    \global\newwrite\@gls@deffile
    \immediate\openout\@gls@deffile=\jobname.glsdefs
  }%
  {%
    \ifglsentryexists{#1}%
    {%
      \xifinlist{#1}{\gls@docdef@labellist}%
      {%
        \GlossariesWarning{Entry ‘#1’ has already been defined in the
          document environment (ignoring new definition)}%
      }%
      {%
        \xifinlist{#1}{\gls@glsdefs@labellist}%
        {%
          \gls@defglossaryentry{#1}{#2}%
          \gls@docdef@addlabel{#1}%
          \@gls@writedef{#1}%
        }%
        {%
          \GlossariesWarning{Entry ‘#1’ has already been defined in
            the preamble (in which case remove the duplicate definition)
            or you have just updated to glossaries.sty
            v4.47 (in which case delete ‘\jobname.glsdefs’)}%
        }%
      }%
    }%
  }%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}

```

```

        \gls@docdef@addlabel{#1}%
        \@gls@writedef{#1}%
    }%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*\gls@begindocdefs{%
    \gls@enablesavenonumberlist
    \edef\@gls@restreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
    \makeatletter
    \InputIfFileExists{\jobname.glsdefs}{-}{-}%
    \@gls@restreat
    \undef\@gls@restreat
    \gls@defdocnewglossaryentry
}
\newcommand*\glsdefs@newdocentry}[2]{%
    \ifglsentryexists{#1}%
    {\GlossariesWarning{entry ‘#1’ has already been defined in the preamble
        (if you have moved the definition there from the document
        environment, delete \jobname.glsdefs)}}%
    {%
        \gls@glsdefs@addlabel{#1}%
        \gls@defglossaryentry{#1}{#2}%
    }%
}
\newcommand*\@gls@writedef}[1]{%
    \immediate\write\@gls@deffile
    {%
        \string\glsdefs@newdocentry{\glsdetoklabel{#1}}\glspercentchar^^J%
        \expandafter\@gobble\string\{\glspercentchar%
    }%
    \@for\@gls@map:=\@gls@keymap\do
    {%
        \letcs\glo@value{glo@\glsdetoklabel{#1}}\expandafter\@secondoftwo\@gls@map}%
        \ifdef\glo@value
        {%
            \@onelevel@sanitize\glo@value
            \immediate\write\@gls@deffile
            {%
                \expandafter\@firstoftwo\@gls@map
                =\expandafter\@gobble\string\{\glo@value\expandafter\@gobble\string\},%
                \glspercentchar
            }%
        }%
    }%
}
\gls@writedefhook
\immediate\write\@gls@deffile
{%
    \glspercentchar^^J%
}

```



```

    \expandafter\@gobble\string\}\glspercentchar^^J%
  }%
}
\newcommand*{\@gls@keymap}{%
  {name}{name},%
  {sort}{sortvalue},% unescaped sort value
  {type}{type},%
  {first}{first},%
  {firstplural}{firstpl},%
  {text}{text},%
  {plural}{plural},%
  {description}{desc},%
  {descriptionplural}{descplural},%
  {symbol}{symbol},%
  {symbolplural}{symbolplural},%
  {user1}{useri},%
  {user2}{userii},%
  {user3}{useriii},%
  {user4}{useriv},%
  {user5}{userv},%
  {user6}{uservi},%
  {long}{long},%
  {longplural}{longpl},%
  {short}{short},%
  {shortplural}{shortpl},%
  {counter}{counter},%
  {parent}{parent}%
}
\newcommand*{\@gls@fetchfield}[2]{%
  \edef\@gls@thisval{#2}%
  \@for\@gls@map:=\@gls@keymap\do{%
    \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
    \ifdefequal{\@this@key}{\@gls@thisval}%
    {%
      \edef#1{\expandafter\@secondoftwo\@gls@map}%
      \@endfortrue
    }%
  }%
}%
}
\newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
\newcommand*{\@sglsaddstoragekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
}%
}

```

```

\@glsaddstoragekey{#1}%
}
\newcommand*{\@glsaddstoragekey}[3]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{\@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
}%
{%
\PackageError{glossaries}{Key ‘#1’ already exists}{}%
}%
}
\newcommand*{\glsaddkey}{\@ifstar\@sglsaddkey\@glsaddkey}
\newcommand*{\@sglsaddkey}[1]{%
\key@ifundefined{glossentry}{#1}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname gls@assign@#1@field\endcsname}[2]{%
\@gls@expand@field{##1}{#1}{##2}%
}%
}%
{}%
\@glsaddkey{#1}%
}
\newcommand*{\@glsaddkey}[7]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{\@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
\newcommand*{#4}[1]{\@Gls@entry@field{##1}{#1}}%
\ifcsdef{@gls@user@#1@}%
{%
\PackageError{glossaries}%
{Can’t define ‘\string#5’ as helper command
‘\expandafter\string\csname @gls@user@#1@\endcsname’ already exists}%
{}%
}%
{}%
}

```

```

\expandafter\newcommand\expandafter*\expandafter
  {\csname @gls@user@#1\endcsname}[2][]{%
    \new@ifnextchar[%
      {\csuse{@gls@user@#1@}{##1}{##2}}%
      {\csuse{@gls@user@#1@}{##1}{##2}[]}}%
\csdef{@gls@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{#3{##2}##3}%
}%
\newrobustcmd*{#5}{%
  \expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
}%
\ifcsdef{@Gls@user@#1@}{%
  {%
    \PackageError{glossaries}{%
      {Can't define '\string#6' as helper command
        '\expandafter\string\csname @Gls@user@#1\endcsname' already exists}}%
    }%
  }%
  {%
    \expandafter\newcommand\expandafter*\expandafter
      {\csname @Gls@user@#1\endcsname}[2][]{%
        \new@ifnextchar[%
          {\csuse{@Gls@user@#1@}{##1}{##2}}%
          {\csuse{@Gls@user@#1@}{##1}{##2}[]}}%
\csdef{@Gls@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{#4{##2}##3}%
}%
\newrobustcmd*{#6}{%
  \expandafter\@gls@hyp@opt\csname @Gls@user@#1\endcsname}%
}%
\ifcsdef{@GLS@user@#1@}{%
  {%
    \PackageError{glossaries}{%
      {Can't define '\string#7' as helper command
        '\expandafter\string\csname @GLS@user@#1\endcsname' already exists}}%
    }%
  }%
  {%
    \expandafter\newcommand\expandafter*\expandafter
      {\csname @GLS@user@#1\endcsname}[2][]{%
        \new@ifnextchar[%
          {\csuse{@GLS@user@#1@}{##1}{##2}}%
          {\csuse{@GLS@user@#1@}{##1}{##2}[]}}%
\csdef{@GLS@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{\mfirstucMakeUppercase{#3{##2}##3}}%
}%
\newrobustcmd*{#7}{%
  \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
}%
}%

```

```

    {%
      \PackageError{glossaries}{Key ‘#1’ already exists}{}%
    }%
  }
\newcommand{\glsfieldxdef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\glo@label @#2}%
    {%
      \protected@csxdef{glo@\glo@label @#2}{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\glsfielddedef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\glo@label @#2}%
    {%
      \protected@csedef{glo@\glo@label @#2}{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\glsfieldgdef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\glo@label @#2}%
    {%
      \expandafter\gdef\csname glo@\glo@label @#2\endcsname{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\glsfieldddef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\glo@label @#2}%
    {%
      \expandafter\def\csname glo@\glo@label @#2\endcsname{#3}%
    }%
  }%
}

```

```

}%
{%
  \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
}%
}%
}
\newcommand{\glsfieldfetch}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \letcs#3{glo@\@glo@label @#2}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand*{\glsunexpandedfieldvalue}[2]{%
  \ifcsname glo@\glsdetoklabel{#1}@#2\endcsname
  \expandafter\expandonce\csname glo@\glsdetoklabel{#1}@#2\expandafter\endcsname
  \fi
}
\newcommand{\ifglsfieldeq}[5]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand{\ifglsfielddfeq}[5]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \expandafter\ifdefstrequal
    \csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}%
}

```

```

}
\newcommand{\ifglsfieldcseq}[5]{%
\glsdoifexists{#1}%
{%
\protected@edef\glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\glo@label @#2}%
{%
\ifcsstrequal{glo@\glo@label @#2}{#3}{#4}{#5}%
}%
{%
\PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
}%
}%
}
\newcommand*{\glswritedefhook}{}
\newcommand*{\gls@assign@desc}[1]{%
\gls@assign@field{#1}{desc}{\glo@desc}%
\gls@assign@field{\glo@desc}{#1}{descplural}{\glo@descplural}%
}
\newcommand{\longnewglossaryentry}[3]{%
\glsdoifnoexists{#1}%
{%
\bgrou
\let\org@newglossaryentryprehook\@newglossaryentryprehook
\long\def\@newglossaryentryprehook{%
\long\def\glo@desc{#3\leavevmode\unskip\nopostdesc}%
\org@newglossaryentryprehook
}%
\renewcommand*{\gls@assign@desc}[1]{%
\global\cslet{glo@\glsdetoklabel{#1}@desc}{\glo@desc}%
\global\cslet{glo@\glsdetoklabel{#1}@descplural}{\glo@desc}%
}
\gls@defglossaryentry{#1}{#2}%
\egrou
}
}
\@onlypreamble{\longnewglossaryentry}
\newcommand{\longprovideglossaryentry}[3]{%
\ifglsentryexists{#1}{}%
{\longnewglossaryentry{#1}{#2}{#3}}%
}
\@onlypreamble{\longprovideglossaryentry}
\newcommand{\gls@defglossaryentry}[2]{%
\let\GlsSetQuote\gls@nosetquote
\protected@edef\glo@label{\glsdetoklabel{#1}}%
\let\glslabel\glo@label
\let\glo@name\gls@name
\let\glo@desc\gls@desc
\let\glo@descplural\gls@default@value
\let\glo@type\gls@default@value

```

```

\let@glo@symbol@gls@default@value
\let@glo@symbolplural@gls@default@value
\let@glo@text@gls@default@value
\let@glo@plural@gls@default@value
\let@glo@first@gls@default@value
\let@glo@firstplural@gls@default@value
\let@glo@sort@gls@default@value
\let@glo@counter@gls@default@value
\def@glo@see{}%
\def@glo@parent{}%
\def@glo@prefix{}%
@gls@initnonnumberlist
\def@glo@useri{}%
\def@glo@userii{}%
\def@glo@useriii{}%
\def@glo@useriv{}%
\def@glo@userv{}%
\def@glo@uservi{}%
\def@glo@short{}%
\def@glo@shortpl{}%
\def@glo@long{}%
\def@glo@longpl{}%
@newglossaryentryprehook
\setkeys{glossentry}{#2}%
\ifundef@glsdefaulttype
{%
  \PackageError{glossaries}%
  {No default glossary type (have you used ‘nomain’ by mistake?)}%
  {If you use package option ‘nomain’ you must define
   a new glossary before you can define entries}%
}%
{}%
\gls@assign@field{\glsdefaulttype}{\glo@label}{type}{\glo@type}%
\protected@edef@glo@type{\glsentrytype{\glo@label}}%
\ifcsundef{glo@list@\glo@type}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘\glo@type’ has not been defined}%
  {You need to define a new glossary type, before making entries
   in it}%
}%
{}%
\ifignoredglossary@glo@type
{%
  \ifx@glo@desc@glsnodesc
    \let@glo@desc@empty
  \fi
}%
{}%
{}%

```

```

\protected@edef\glo@list@{\csname glo@list@\@glo@type\endcsname}%
\expandafter\protected@xdef\csname glo@list@\@glo@type\endcsname{%
  \glo@list@{\@glo@label},}%
}%
\gls@level=0\relax
\ifx\@glo@parent\@empty
  \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
\else
  \ifdefequal\@glo@label\@glo@parent%
  {%
    \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
    \def\@glo@parent{}%
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
  }%
  {%
    \ifglsentryexists{\@glo@parent}%
    {%
      \expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
        \@glo@parent}%
      \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
      \advance\gls@level by 1\relax
      \ifx\@glo@name\@glsnoname
        \expandafter\let\expandafter\@glo@name
        \csname glo@\@glo@parent @name\endcsname
        \ifx\@glo@plural\@gls@default@value
          \expandafter\let\expandafter\@glo@plural
          \csname glo@\@glo@parent @plural\endcsname
        \fi
      \fi
    }%
  }%
  {%
    \PackageError{glossaries}%
    {%
      Invalid parent ‘\@glo@parent’
      for entry ‘\@glo@label’ - parent doesn’t exist%
    }%
  }%
  {%
    Parent entries must be defined before their children%
  }%
  \def\@glo@parent{}%
  \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
}
}%
\fi
\expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%
\gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
\letcs\@glo@sort{glo@\@glo@label @sortvalue}%
\gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @text\endcsname\glspluralsuffix}%

```



```

        {\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @text\endcsname}%
    {\@glo@label}{first}{\@glo@first}%
\ifx\@glo@first\@gls@default@value
    \expandafter\gls@assign@field\expandafter
        {\csname glo@\@glo@label @plural\endcsname}%
        {\@glo@label}{firstpl}{\@glo@firstplural}%
\else
    \expandafter\gls@assign@field\expandafter
        {\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
        {\@glo@label}{firstpl}{\@glo@firstplural}%
\fi
\ifcsundef{@glotype@\@glo@type @counter}%
{%
    \def\@glo@defaultcounter{\glscounter}%
}%
{%
\letcs\@glo@defaultcounter{@glotype@\@glo@type @counter}%
}%
\gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
\gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
\gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
\gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
\gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
\gls@assign@field{}{\@glo@label}{userv}{\@glo@userv}%
\gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
\gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
\gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
\gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
\gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%
\ifx\@glo@name\@glsnoname
    \@glsnoname
    \let\@glo@name\@gls@default@value
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%
\ifcsundef{glo@\@glo@label @numberlist}%
{%
    \csxdef{glo@\@glo@label @numberlist}{%
        \noexpand\@gls@missingnumberlist{\@glo@label}}%
}%
{%
}%
\@gls@storenonumberlist{\@glo@label}%
\def\@glo@@desc{\@glo@first}%
\ifx\@glo@desc\@glo@@desc
    \let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@glsnodesc
    \@glsnodesc
    \let\@glo@desc\@gls@default@value

```

```

\fi
\gls@assign@desc{\@glo@label}%
\@gls@defsort{\@glo@type}{\@glo@label}%
\def\@glo@@symbol{\@glo@text}%
\ifx\@glo@symbol\@glo@@symbol
  \let\@glo@symbol\@glo@text
\fi
\gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
\expandafter
  \gls@assign@field\expandafter
  {\csname glo@\@glo@label @symbol\endcsname}
  {\@glo@label}{symbolplural}{\@glo@symbolplural}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagfalse\endcsname{%
  \noexpand\global
  \noexpand\let\expandafter\noexpand
  \csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagtrue\endcsname{%
  \noexpand\global
  \noexpand\let\expandafter\noexpand
  \csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
}%
\csname glo@\@glo@label @flagfalse\endcsname
\@glo@autosee
\ifignoredglossary\@glo@type
{%
  \csdef{glo@\@glo@label @index}{}%
}
{%
  \do@glo@storeentry{\@glo@label}%
}%
\@newglossaryentry@defcounters
\@newglossaryentryposthook
}
\newcommand*{\@glo@autosee}{%
  \ifdefvoid\@glo@see{%
    {%
      \protected@edef\@do@glssee{%
        \noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
        \noexpand\expandafter\noexpand\@glssee\noexpand\@glo@list{\@glo@label}}%
      \do@glssee
    }%
    \@glo@autoseehook
  }%
}
\newcommand*{\@glo@autoseehook}{}
\newcommand*{\@newglossaryentryprehook}{}
\newcommand*{\@newglossaryentryposthook}{}
\newcommand*{\@newglossaryentry@defcounters}{}
\newcommand*{\glsmoveentry}[2]{%
  \protected@edef\@glo@thislabel{\glsdetoklabel{#1}}%

```



```

\else
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
\csname glo@\glo@parent @index\endcsname\@gls@levelchar
\@glo@sort\@gls@actualchar\@glo@prefix
\@glossarysubentryfield
{\csname glo@#1@level\endcsname}\@glo@esclabel}%
}%
\fi
\fi
}
\AtBeginDocument{%
\@ifpackageloaded{amsmath}%
{\let\gls@ifnotmeasuring\@gls@ifnotmeasuring}%
{}}%
}
\newcommand*{\@gls@ifnotmeasuring}[1]{%
\ifmeasuring@
\else
#1%
\fi
}
\newcommand*\gls@ifnotmeasuring[1]{#1}
\def\@gls@patchtabularx#1\hbox#2#3!!{%
\def\TX@trial##1{#1\hbox{\let\glsunset\@gobble#2}#3}%
}
\newcommand*\glspatchtabularx{%
\ifdef\TX@trial
{%
\expandafter\@gls@patchtabularx\TX@trial{##1}!!%
\let\glspatchtabularx\relax
}%
{}}%
}
\newcommand*{\glsreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glsreset{#1}%
}%
}%
}
\newcommand*{\glslocalreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glslocalreset{#1}%
}%
}%
}

```

```

}
\newcommand*\glsunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glsunset{#1}%
    }%
  }%
}
\newcommand*\glslocalunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glslocalunset{#1}%
    }%
  }%
}
\newcommand*\@glslocalunset{\@glslocalunset}
\newcommand*\@glslocalunset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
}
\newcommand*\@glsunset{\@glsunset}
\newcommand*\@glsunset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
}
\newcommand*\@glslocalreset{\@glslocalreset}
\newcommand*\@glslocalreset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
}
\newcommand*\@glsreset{\@glsreset}
\newcommand*\@glsreset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
}
\newcommand*\glsresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsreset{\@glsentry}%
  }%
}
\newcommand*\glslocalresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalreset{\@glsentry}%
  }%
}
\newcommand*\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%

```

```

    \glsunset{\@glsentry}%
  }%
}
\newcommand*\glslocalunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalunset{\@glsentry}%
  }%
}
\newcommand*\@newglossaryentry@defcounters{%
  \csdef{glo@\@glo@label @currcount}{0}%
  \csdef{glo@\@glo@label @prevcount}{0}%
}
\newcommand*\glsenableentrycount{%
  \let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
  \renewcommand*\gls@defdocnewglossaryentry{%
    \renewcommand*\newglossaryentry[2]{%
      \PackageError{glossaries}{\string\newglossaryentry\space
        may only be used in the preamble when entry counting has
        been activated}{If you use \string\glsenableentrycount\space
        you must place all entry definitions in the preamble not in
        the document environment}%
    }%
  }%
}
\newcommand*\glsentrycurrcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
  {0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*\glsentryprevcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
  {0}{\@gls@entry@field{##1}{prevcount}}%
}%
\renewcommand*\@glsunset}[1]{%
  \@glsunset{##1}%
  \@gls@increment@currcount{##1}%
}%
\renewcommand*\@glslocalunset}[1]{%
  \@glslocalunset{##1}%
  \@gls@local@increment@currcount{##1}%
}%
\renewcommand*\@glsreset}[1]{%
  \@glsreset{##1}%
  \csgdef{glo@\glsdetoklabel{##1}@currcount}{0}%
}%
\renewcommand*\@glslocalreset}[1]{%
  \@glslocalreset{##1}%
  \csdef{glo@\glsdetoklabel{##1}@currcount}{0}%
}%
\def\@cgls###1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax

```

```

        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \@gls@{##1}{##2}[##3]%
    \fi
}%
\def\@cglsp@##1##2[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \@cglsp@{##1}{##2}[##3]%
    \fi
}%
\def\@Gls@##1##2[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \@Gls@{##1}{##2}[##3]%
    \fi
}%
\def\@Glspl@##1##2[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \@Glspl@{##1}{##2}[##3]%
    \fi
}%
}
\AtEndDocument{\@gls@write@entrycounts}%
\renewcommand*{\@gls@entry@count}[2]{%
    \csdef{glo@\glsdetoklabel{##1}@prevcount}{##2}%
}%
\let\glsetenableentrycount\relax
}
\@onlypreamble\glsetenableentrycount
\newcommand*{\@gls@increment@currcount}[1]{%
    \csxdef{glo@\glsdetoklabel{##1}@currcount}{%
        \number\numexpr\glsetentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@local@increment@currcount}[1]{%
    \csedef{glo@\glsdetoklabel{##1}@currcount}{%
        \number\numexpr\glsetentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@write@entrycounts}{%
    \immediate\write\@auxout
        {\string\providecommand*{\string\@gls@entry@count}[2]{}}%
    \forallglsentries{\@glsentry}{%
        \ifglsused{\@glsentry}%

```

```

        {\immediate\write\@auxout
         {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}%
        }%
    }%
}
\newcommand*{\@gls@entry@count}[2]{%
\newrobustcmd*{\cgl}{\@gls@hyp@opt\@cgl}
\newcommand*{\@cgl}[2][]{%
  \new@ifnextchar[{\@cgl@#1}{#2}]{\@cgl@#1}{#2}[]}%
}
\def\@cgl@#1#2[#3]{%
  \GlossariesWarning{\string\cgl\space is defaulting to
    \string\gls\space since you haven't enabled entry counting}%
  \@cgl@#1}{#2}[#3]%
}
\newcommand*{\cglformat}[2]{%
  \ifglshaslong{#1}{\glsentrylong{#1}}{\glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cGls}{\@gls@hyp@opt\@cGls}
\newcommand*{\@cGls}[2][]{%
  \new@ifnextchar[{\@cGls@#1}{#2}]{\@cGls@#1}{#2}[]}%
}
\def\@cGls@#1#2[#3]{%
  \GlossariesWarning{\string\cGls\space is defaulting to
    \string\Gls\space since you haven't enabled entry counting}%
  \@cGls@#1}{#2}[#3]%
}
\newcommand*{\cGlsformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cglsp}{\@gls@hyp@opt\@cglsp}
\newcommand*{\@cglsp}[2][]{%
  \new@ifnextchar[{\@cglsp@#1}{#2}]{\@cglsp@#1}{#2}[]}%
}
\def\@cglsp@#1#2[#3]{%
  \GlossariesWarning{\string\cglsp\space is defaulting to
    \string\glsp\space since you haven't enabled entry counting}%
  \@cglsp@#1}{#2}[#3]%
}
\newcommand*{\cglspformat}[2]{%
  \ifglshaslong{#1}{\glsentrylongpl{#1}}{\glsentryfirstplural{#1}}#2%
}
\newrobustcmd*{\cGlspl}{\@gls@hyp@opt\@cGlspl}
\newcommand*{\@cGlspl}[2][]{%
  \new@ifnextchar[{\@cGlspl@#1}{#2}]{\@cGlspl@#1}{#2}[]}%
}
\def\@cGlspl@#1#2[#3]{%
  \GlossariesWarning{\string\cGlspl\space is defaulting to
    \string\Glspl\space since you haven't enabled entry counting}%
  \@cGlspl@#1}{#2}[#3]%
}

```



```

}
\newcommand*{\cGlsplformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
}
\newcommand*{\loadglsentries}[2][\@gls@default]{%
  \let\@gls@default\glsdefaulttype
  \def\glsdefaulttype{#1}\input{#2}%
  \let\glsdefaulttype\@gls@default
}
\@onlypreamble{\loadglsentries}
\newcommand*{\glstextformat}[1]{#1}
\newcommand*{\glsentryfmt}{%
  \@gls@default@entryfmt\glsdisplayfirst\glsdisplay
}
\newcommand*{\@@gls@default@entryfmt}[2]{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          #2{\glsentryplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
        }%
        {%
          #1{\glsentryfirstplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
          }%
        }%
      }%
    }%
  }%
  \ifglsused\glslabel
  {%
    \ifbool{glscompatible-3.07}%
    {%
      \protected@edef\@glo@etext{%
        #2{\glsentryplural{\glslabel}}%
          {\glsentrydescplural{\glslabel}}%
          {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
      \xmakefirstuc\@glo@etext
    }%
    {%
      #2{\Glsentryplural{\glslabel}}%
        {\glsentrydescplural{\glslabel}}%
        {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
    }%
  }%
}

```

```

{%
  \ifbool{glscompatible-3.07}%
  {%
    \protected@edef\@glo@etext{%
      #1{\glsentryfirstplural{\glslabel}}%
      {\glsentrydescplural{\glslabel}}%
      {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
    \xmakefirstuc\@glo@etext
  }%
  {%
    #1{\Glsentryfirstplural{\glslabel}}%
    {\glsentrydescplural{\glslabel}}%
    {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
  }%
}%
}%
{%
  \ifglsused\glslabel
  {%
    \mfirstucMakeUppercase{#2{\glsentryplural{\glslabel}}%
      {\glsentrydescplural{\glslabel}}%
      {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
  }%
  {%
    \mfirstucMakeUppercase{#1{\glsentryfirstplural{\glslabel}}%
      {\glsentrydescplural{\glslabel}}%
      {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
  }%
}%
}%
{%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      #2{\glsentrytext{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
    {%
      #1{\glsentryfirst{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
  }%
}%
{%
  \ifglsused\glslabel
  {%
    \ifbool{glscompatible-3.07}%
    {%

```



```

    {%
      #1{\glscustomtext}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{}}%
    }%
  }%
}
\newcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          \glsentryplural{\glslabel}\glsinsert
        }%
        {%
          \glsentryfirstplural{\glslabel}\glsinsert
        }%
      }%
    }%
    \ifglsused\glslabel
    {%
      \Glsentryplural{\glslabel}\glsinsert
    }%
    {%
      \Glsentryfirstplural{\glslabel}\glsinsert
    }%
  }%
  {%
    \ifglsused\glslabel
    {%
      \mfirstucMakeUppercase
      {\glsentryplural{\glslabel}\glsinsert}%
    }%
    {%
      \mfirstucMakeUppercase
      {\glsentryfirstplural{\glslabel}\glsinsert}%
    }%
  }%
}
  }%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      \glsentrytext{\glslabel}\glsinsert
    }%
  }%
}

```

```

        {%
        \glsentryfirst{\glslabel}\glsinsert
        }%
    }%
    {%
    \ifglsused\glslabel
    {%
    \Glsentrytext{\glslabel}\glsinsert
    }%
    {%
    \Glsentryfirst{\glslabel}\glsinsert
    }%
    }%
    {%
    \ifglsused\glslabel
    {%
    \mfirstucMakeUppercase{\glsentrytext{\glslabel}\glsinsert}%
    }%
    {%
    \mfirstucMakeUppercase{\glsentryfirst{\glslabel}\glsinsert}%
    }%
    }%
    }%
    }%
    {%
    \glscustomtext\glsinsert
    }%
}
\newcommand*{\glsgenacfmt}{%
\ifdefempty\glscustomtext
{%
\ifglsused\glslabel
{%
\glsifplural
{%
\glscapscase
{%
\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert
}%
{%
\acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
}%
{%
\mfirstucMakeUppercase
{\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert}%
}%
}%
}%
\glscapscase
{%

```



```

\protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
\xmakefirstuc\gls@text
}
\newcommand*\genplacrfullformat}[2]{%
  \glsentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glsentryshortpl{#1}})}%
}
\newcommand*\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \xmakefirstuc\gls@text
}
\newcommand*\glsdisplayfirst}[4]{#1#4}
\newcommand*\glsdisplay}[4]{#1#4}
\newcommand*\defglsdisplay}[2][\glsdefaulttype]{%
  \GlossariesWarning{\string\defglsdisplay\space is now obsolete.^^J
  Use \string\defglsentryfmt\space instead}%
  \expandafter\def\csname gls@#1@display\endcsname##1##2##3##4{#2}%
  \edef\@gls@doentrydef{%
    \noexpand\defglsentryfmt[#1]{%
      \noexpand\ifcsdef{gls@#1@displayfirst}}%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\glsdisplayfirst}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
    }%
  }%
  \@gls@doentrydef
}
\newcommand*\defglsdisplayfirst}[2][\glsdefaulttype]{%
  \GlossariesWarning{\string\defglsdisplayfirst\space is now obsolete.^^J
  Use \string\defglsentryfmt\space instead}%
  \expandafter\def\csname gls@#1@displayfirst\endcsname##1##2##3##4{#2}%
  \edef\@gls@doentrydef{%
    \noexpand\defglsentryfmt[#1]{%
      \noexpand\ifcsdef{gls@#1@display}}%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\glsdisplay}%
      }%
    }%
  }%
  \@gls@doentrydef
}

```

```

    }%
  }%
}%
\@gls@doentrydef
}
\define@key{glslink}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
}%
{%
  \def\@gls@counter{#1}%
}%
}
\define@key{glslink}{format}{%
  \def\@glsnumberformat{#1}}
\define@boolkey{glslink}{hyper}[true]{}
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\define@boolkey{glslink}{local}[true]{}
\newcommand*{\glslinkvar}[3]{#1}
\newcommand*{\glsifhyper}[2]{%
  \glslinkvar{#1}{#2}{#1}%
  \GlossariesWarning{\string\glsifhyper\space is deprecated. Did
  you mean \string\glsifhyperon\space or \string\glslinkvar?}%
}
\newcommand*{\@gls@hyp@opt}[1]{%
  \let\glslinkvar\@firstofthree
  \let\@gls@hyp@opt@cs#1\relax
  \@ifstar{\s@gls@hyp@opt}%
  {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*{\s@gls@hyp@opt}[1] []{%
  \let\glslinkvar\@secondofthree
  \@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1] []{%
  \let\glslinkvar\@thirdofthree
  \@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
  \@gls@hyp@opt\@gls@@link
}
\newcommand*{\@gls@@link}[3] []{%
  \glsdoifexistsordo{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \@gls@link[#1]{#2}{#3}%
  }
}

```



```

    }{%
      \glstextformat{#3}%
    }%
  \glspostlinkhook
}
\newcommand*{\glspostlinkhook}{}
\newcommand*{@gls@link@checkfirsthyper}{%
  \ifglsused{glslabel}%
  {%
  }%
  {%
    \gls@checkisacronymlist\glstype
    \ifglshyperfirst
      \ifglsisacronymlist
        \ifglsacrfootnote
          \KV@glslink@hyperfalse
        \fi
      \fi
    \else
      \KV@glslink@hyperfalse
    \fi
  }%
  \glslinkcheckfirsthyperhook
}
\newcommand*{\glslinkcheckfirsthyperhook}{}
\newcommand*{\glslinkpostsetkeys}{}
\newcommand{\glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}
\newcommand*{\do@glsdisablehyperinlist}{%
  \expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
  {\KV@glslink@hyperfalse}{}%
}
\newcommand*{@gls@setdefault@glslink@opts}{}
\def\@gls@link[#1]#2#3{%
  \leavevmode
  \protected@edef\glslabel{\glsdetoklabel{#2}}%
  \def\@gls@link@opts{#1}%
  \let\@gls@link@label\glslabel
  \def\@glsnumberformat{glsnumberformat}%
  \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  \protected@edef\glstype{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
  \@gls@setdefault@glslink@opts
  \do@glsdisablehyperinlist
  \do@gls@link@checkfirsthyper
  \setkeys{glslink}{#1}%
  \glslinkpostsetkeys
  \@gls@saveentrycounter
  \@gls@setsort{glslabel}%
  \@do@wrglossary{#2}%
  \ifKV@glslink@hyper

```

```

        \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
    \else
        \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
    \fi
    \let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
}
\newcommand*{\glolinkprefix}{glo:}
\def\glsentrycounter{\glscounter}%
\newcommand*{\@gls@saveentrycounter}{%
    \def\@gls@Hcounter{}%
    \ifthenelse{\equal{\@gls@counter}{equation}}%
    {
        \ifcsundef{xatlevel@}%
        {%
            \edef\theglsentrycounter{\expandafter\noexpand
                \csname the\@gls@counter\endcsname}%
        }%
        {%
            \ifx\xatlevel@\@empty
                \edef\theglsentrycounter{\expandafter\noexpand
                    \csname the\@gls@counter\endcsname}%
            \else
                \savecounters@
                \advance\c@equation by 1\relax
                \edef\theglsentrycounter{\csname the\@gls@counter\endcsname}%
                \ifcsundef{theH\@gls@counter}%
                {%
                    \def\@gls@Hcounter{\theglsentrycounter}%
                }%
                {%
                    \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
                }%
                \protected@edef\theHglsentrycounter{\@gls@Hcounter}%
                \restorecounters@
            \fi
        }%
    }%
    {%
        \edef\theglsentrycounter{\expandafter\noexpand
            \csname the\@gls@counter\endcsname}%
    }%
    \ifx\@gls@Hcounter\@empty
        \ifcsundef{theH\@gls@counter}%
        {%
            \def\theHglsentrycounter{\theglsentrycounter}%
        }%
        {%
            \protected@edef\theHglsentrycounter{\expandafter\noexpand
                \csname theH\@gls@counter\endcsname}%
        }%
    }%
}

```

```

\fi
}
\def\@set@glo@numformat#1#2#3#4{%
\expandafter\@glo@check@mkidxrangechar#3\@nil
\protected@edef#1{%
\@glo@prefix setentrycounter[#4]{#2}%
\expandafter\string\csname\@glo@suffix\endcsname
}%
\@gls@checkmkidxchars#1%
}
\def\@glo@check@mkidxrangechar#1#2\@nil{%
\if#1\relax
\def\@glo@prefix{()}%
\if\relax#2\relax
\def\@glo@suffix{glsnumberformat}%
\else
\def\@glo@suffix{#2}%
\fi
\else
\if#1\relax
\def\@glo@prefix{})%
\if\relax#2\relax
\def\@glo@suffix{glsnumberformat}%
\else
\def\@glo@suffix{#2}%
\fi
\else
\def\@glo@prefix{}\def\@glo@suffix{#1#2}%
\fi
\fi}
\newcommand*{\@gls@escbsdq}[1]{%
\def\@gls@checkedmkidx{}%
\let\gls@xdystring=#1\relax
\@onelevel@sanitize\gls@xdystring
\edef\do@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null}%
\do@gls@xdycheckbackslash
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\@for\@gls@tmp:=\gls@protected@pagefmts\do
{%
\edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\ \expandonce\@gls@tmp}%
\@onelevel@sanitize\@gls@sanitized@tmp
\edef\gls@dostubst{%
\noexpand\DTLsubstituteall\noexpand\gls@xdystring
{\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
}%
}

```

```

    \gls@dostsubst
  }%
  \let#1=\gls@xdyststring
}
\newcommand{\@gls@checkmkidxchars}[1]{%
  \ifglsxindy
    \@gls@escbsdq{#1}%
  \else
    \def\@gls@checkedmkidx{%
      \expandafter\@gls@checkquote#1\@nil""\null
      \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
      \def\@gls@checkedmkidx{%
        \expandafter\@gls@checkescquote#1\@nil""\null
        \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
        \def\@gls@checkedmkidx{%
          \expandafter\@gls@checkescactual#1\@nil"??\null
          \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
          \def\@gls@checkedmkidx{%
            \expandafter\@gls@checkactual#1\@nil??\null
            \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
            \def\@gls@checkedmkidx{%
              \expandafter\@gls@checkbar#1\@nil||\null
              \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
              \def\@gls@checkedmkidx{%
                \expandafter\@gls@checkescbar#1\@nil|||\null
                \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                \def\@gls@checkedmkidx{%
                  \expandafter\@gls@checklevel#1\@nil!!\null
                  \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                  \def\@gls@checkedmkidx{%
                    \expandafter\@gls@checkesclevel#1\@nil!!\null
                    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                    \def\@gls@checkedmkidx{%
                      \edef\@do@gls@checkesc{%
                        \noexpand\@gls@checkesc\expandonce{#1}\noexpand\@nil
                        \@backslashchar\@backslashchar\noexpand\@nil
                      }%
                      \@do@gls@checkesc
                      \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                    \fi
                  \def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
                  \newtoks\@gls@tmpb
                  \def\@gls@checkquote#1"#2"#3\@nil{%
                    \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
                    \toks@=#1}%
                    \ifx\@nil#2\@nil
                    \ifx\@nil#3\@nil
                    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
                    \def\@gls@checkquote{\relax}%
                  \def\@gls@checkquote{\relax}%
                \fi
              \fi
            \fi
          \fi
        \fi
      \fi
    }%
  \fi
}

```

```

\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
\def\@@gls@checkquote{\@gls@checkquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@quotechar}%
\ifx\null#3\null
\def\@@gls@checkquote{\@gls@checkquote#2""\null}%
\else
\def\@@gls@checkquote{\@gls@checkquote#2"#3\null}%
\fi
\fi
\@@gls@checkquote
}
\def\@gls@checkescquote#1"#2"#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkescquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar
\@gls@quotechar\string"\@gls@quotechar}%
\def\@@gls@checkescquote{\@gls@checkescquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar}%
\ifx\null#3\null
\def\@@gls@checkescquote{\@gls@checkescquote#2""\null}%
\else
\def\@@gls@checkescquote{\@gls@checkescquote#2"#3\null}%
\fi
\fi
\@@gls@checkescquote
}
\def\@gls@checkescactual#1\?#2\?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkescactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar

```

```

        \@gls@quotechar\string"\@gls@actualchar}%
        \def\@gls@checkescactual{\@gls@checkescactual#3\null}%
    \fi
\else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
    \@gls@quotechar\string"\@gls@actualchar}%
    \ifx\null#3\null
        \def\@gls@checkescactual{\@gls@checkescactual#2\?\?\null}%
    \else
        \def\@gls@checkescactual{\@gls@checkescactual#2\?#3\null}%
    \fi
\fi
\@gls@checkescactual
}
\def\@gls@checkescbar#1\|#2\|#3\null{%
    \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
    \toks@={#1}%
    \ifx\null#2\null
        \ifx\null#3\null
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
            \def\@gls@checkescbar{\relax}%
        \else
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
            \@gls@quotechar\string"\@gls@encapchar
            \@gls@quotechar\string"\@gls@encapchar}%
            \def\@gls@checkescbar{\@gls@checkescbar#3\null}%
        \fi
    \else
        \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\string"\@gls@encapchar}%
        \ifx\null#3\null
            \def\@gls@checkescbar{\@gls@checkescbar#2\|\|\null}%
        \else
            \def\@gls@checkescbar{\@gls@checkescbar#2\|#3\null}%
        \fi
    \fi
\@gls@checkescbar
}
\def\@gls@checkesclevel#1\!#2\!#3\null{%
    \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
    \toks@={#1}%
    \ifx\null#2\null
        \ifx\null#3\null
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
            \def\@gls@checkesclevel{\relax}%
        \else
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
            \@gls@quotechar\string"\@gls@levelchar
            \@gls@quotechar\string"\@gls@levelchar}%
            \def\@gls@checkesclevel{\@gls@checkesclevel#3\null}%
        \fi
    \fi
}

```

```

\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\@"\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!\!\null}%
\else
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!#3\null}%
\fi
\fi
\fi
\@gls@checkesclevel
}
\def\@gls@checkbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@gls@checkbar{\@gls@checkbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkbar{\@gls@checkbar#2|\!\null}%
\else
\def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}
\def\@gls@checklevel#1!#2!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checklevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
\def\@gls@checklevel{\@gls@checklevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar}%

```

```

\ifx\null#3\null
  \def\@gls@checklevel{\@gls@checklevel#2!\null}%
\else
  \def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
\fi
\fi
\@gls@checklevel
}
\def\@gls@checkactual#1?#2?#3\null{%
  \gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkactual{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
      \def\@gls@checkactual{\@gls@checkactual#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@gls@actualchar}%
    \ifx\null#3\null
      \def\@gls@checkactual{\@gls@checkactual#2??\null}%
    \else
      \def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
    \fi
  \fi
  \@gls@checkactual
}
\bgroup
\catcode'\0\relax
\catcode'\12\relax
\gdef\@gls@checkesc#1\#2\#3\null{%
  \gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkesc{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@backslashchar\@gls@quotechar\@backslashchar}%
      \def\@gls@checkesc{\@gls@checkesc#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@backslashchar}%
    \ifx\null#3\null

```



```

        |def|@gls@checkesc{|@gls@checkesc#2\\|null}%
    |else
        |def|@gls@checkesc{|@gls@checkesc#2\#3|null}%
    |fi
|fi
|@gls@checkesc
}
|egroup
\def\@gls@xdycheckquote#1"#2"#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@xdycheckquote{\relax}%
    |else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \string"\string"}%
      \def\@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
    |fi
  |else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \string"}%
    \ifx\null#3\null
      \def\@gls@xdycheckquote{\@gls@xdycheckquote#2"\null}%
    |else
      \def\@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
    |fi
  |fi
  \@gls@xdycheckquote
}
\edef\def@gls@xdycheckbackslash{%
\noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
  ##2\@backslashchar##3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter
  {\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={##1}%
\noexpand\ifx\noexpand\null##2\noexpand\null
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{\relax}%
\noexpand|else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
  \@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
  \noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
\noexpand|fi
\noexpand|else

```

```

\newcommand\gls@checkedmkidx{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
  \@backslashchar\@backslashchar}%
\newcommand\gls@xdycheckbackslash{%
  \noexpand\def\noexpand\@gls@xdycheckbackslash{%
    \noexpand\@gls@xdycheckbackslash##2\@backslashchar
    \@backslashchar\noexpand\null}%
  \noexpand\else
  \noexpand\def\noexpand\@gls@xdycheckbackslash{%
    \noexpand\@gls@xdycheckbackslash##2\@backslashchar
    ##3\noexpand\null}%
  \noexpand\fi
  \noexpand\fi
  \noexpand\@gls@xdycheckbackslash
}%
}
\def\gls@xdycheckbackslash

\newlength\gls@tmplen
\newcommand*\glsdohypertarget}[2]{%
  \gls@showtarget{#1}%
  \settoheight{\gls@tmplen}{#2}%
  \raisebox{\gls@tmplen}{\hypertarget{#1}{}}#2%
}
\newcommand*\glsdohyperlink}[2]{%
  \gls@showtarget{#1}%
  \hyperlink{#1}{#2}%
}
\newcommand*\glsdonohyperlink}[2]{#2}
\ifcsundef{hyperlink}%
{%
  \let\@glslink\glsdonohyperlink
}%
{%
  \let\@glslink\glsdohyperlink
}
\ifcsundef{hypertarget}%
{%
  \let\@glstarget\@secondoftwo
}%
{%
  \let\@glstarget\glsdohypertarget
}
\newcommand*\glsdisablehyper{%
  \KV@glslink@hyperfalse
  \let\@glslink\glsdonohyperlink
  \let\@glstarget\@secondoftwo
}
\newcommand*\glsenablehyper{%
  \KV@glslink@hypertrue

```

```

\let\@glslink\glsdohyperlink
\let\@glstarget\glsdohypertarget
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\newrobustcmd*\gls{\@gls@hyp@opt\@gls}
\newcommand*\@gls}[2] [] {%
  \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2} []}%
}
\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@firstofthree
    \let\gls custom text\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*\@Gls{\@gls@hyp@opt\@Gls}
\newcommand*\@Gls}[2] [] {%
  \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2} []}%
}
\def\@Gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@secondofthree
    \let\gls custom text\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*\@GLS{\@gls@hyp@opt\@GLS}

```

```

\newcommand*{\@GLS}[2] [] {%
  \new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2} []}%
}
\def\@GLS@#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@thirdofthree
    \let\gls custom text\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\glspl}{\@gls@hyp@opt\@glspl}
\newcommand*{\@glspl}[2] [] {%
  \new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2} []}%
}
\def\@glspl@#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\gls caps case\@firstofthree
    \let\gls custom text\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
\newcommand*{\@Glspl}[2] [] {%
  \new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2} []}%
}
\def\@Glspl@#1#2[#3] {%
  \glsdoifexists{#2}%
  {%

```

```

\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@firstoftwo
\let\glsifcaps\@secondofthree
\let\glsifcustomtext\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\ifKV@glslink@local
  \glslocalunset{#2}%
\else
  \glsunset{#2}%
\fi
}%
\glspostlinkhook
}
\newrobustcmd*{\GLSpl}{\@gls@hyp@opt\@GLSpl}
\newcommand*{\@GLSpl}[2][]{%
  \new@ifnextchar[{\@GLSpl@{#1}{#2}}{\@GLSpl@{#1}{#2}[]}%
}
\def\@GLSpl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glsifcaps\@thirdofthree
    \let\glsifcustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\glsdisp}{\@gls@hyp@opt\@glsdisp}
\newcommand*{\@glsdisp}[3][]{%
  \glsdoifexists{#2}{%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glsifcaps\@firstofthree
    \def\glsifcustomtext{#3}%
    \def\glsinsert{}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else

```

```

        \glsunset{#2}%
    \fi
}%
\glspostlinkhook
}
\newcommand*{\@gls@link@nocheckfirsthyper}{}
\newcommand{\@gls@field@link}[3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \@gls@link{#1}{#2}{#3}%
    }%
    \glspostlinkhook
}
\newrobustcmd*{\glsstext}{\@gls@hyp@opt\@glsstext}
\newcommand*{\@glsstext}[2][{}]{%
    \new@ifnextchar[{\@glsstext@{#1}{#2}}{\@glsstext@{#1}{#2}[]]}
\def\@glsstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentrytext{#2}#3}%
}
\newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
\newcommand*{\@GLStext}[2][{}]{%
    \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2}[]]}
\def\@GLStext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrytext{#2}#3}}%
}
\newrobustcmd*{\Glsstext}{\@gls@hyp@opt\@Glsstext}
\newcommand*{\@Glsstext}[2][{}]{%
    \new@ifnextchar[{\@Glsstext@{#1}{#2}}{\@Glsstext@{#1}{#2}[]]}
\def\@Glsstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentrytext{#2}#3}%
}
\newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
\newcommand*{\@glsfirst}[2][{}]{%
    \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2}[]]}
\def\@glsfirst@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
}
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
\newcommand*{\@Glsfirst}[2][{}]{%
    \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2}[]]}
\def\@Glsfirst@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
\newcommand*{\@GLSfirst}[2][{}]{%
    \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2}[]]}
\def\@GLSfirst@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirst{#2}#3}}%
}
}

```

```

\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
\newcommand*{\@glsplural}[2] []{%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} []}]
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\newcommand*{\@Glsplural}[2] []{%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} []}]
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
\newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
\newcommand*{\@GLSplural}[2] []{%
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} []}]
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryplural{#2}#3}}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{\@glsfirstplural}[2] []{%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} []}]
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
\newcommand*{\@Glsfirstplural}[2] []{%
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2} []}]
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
\newcommand*{\@GLSfirstplural}[2] []{%
  \new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} []}]
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirstplural{#2}#3}}%
}
\newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
\newcommand*{\@glsname}[2] []{%
  \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2} []}]
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryname{#2}#3}%
}
\newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}
\newcommand*{\@Glsname}[2] []{%
  \new@ifnextchar[{\@Glsname@{#1}{#2}}{\@Glsname@{#1}{#2} []}]
\def\@Glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryname{#2}#3}%
}
\newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}
\newcommand*{\@GLSname}[2] []{%

```

```

    \new@ifnextchar [{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}]
\def\@GLSname@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryname{#2}#3}}%
}
\newrobustcmd*{\glsdesc}{\@gls@hyp@opt\@glsdesc}
\newcommand*{\@glsdesc}[2] []{%
    \new@ifnextchar [{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2} []}]
\def\@glsdesc@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentrydesc{#2}#3}%
}
\newrobustcmd*{\Glsdesc}{\@gls@hyp@opt\@Glsdesc}
\newcommand*{\@Glsdesc}[2] []{%
    \new@ifnextchar [{\@Glsdesc@{#1}{#2}}{\@Glsdesc@{#1}{#2} []}]
\def\@Glsdesc@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%
}
\newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\newcommand*{\@GLSdesc}[2] []{%
    \new@ifnextchar [{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}]
\def\@GLSdesc@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydesc{#2}#3}}%
}
\newrobustcmd*{\glsdescplural}{\@gls@hyp@opt\@glsdescplural}
\newcommand*{\@glsdescplural}[2] []{%
    \new@ifnextchar [{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}]
\def\@glsdescplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\Glsdescplural}{\@gls@hyp@opt\@Glsdescplural}
\newcommand*{\@Glsdescplural}[2] []{%
    \new@ifnextchar [{\@Glsdescplural@{#1}{#2}}{\@Glsdescplural@{#1}{#2} []}]
\def\@Glsdescplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\newcommand*{\@GLSdescplural}[2] []{%
    \new@ifnextchar [{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}]
\def\@GLSdescplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydescplural{#2}#3}}%
}
\newrobustcmd*{\glssymbol}{\@gls@hyp@opt\@glssymbol}
\newcommand*{\@glssymbol}[2] []{%
    \new@ifnextchar [{\@glssymbol@{#1}{#2}}{\@glssymbol@{#1}{#2} []}]
\def\@glssymbol@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\Glsymbol}{\@gls@hyp@opt\@Glsymbol}
\newcommand*{\@Glsymbol}[2] []{%
    \new@ifnextchar [{\@Glsymbol@{#1}{#2}}{\@Glsymbol@{#1}{#2} []}]
\def\@Glsymbol@#1#2[#3]{%

```



```

    \@gls@field@link{#1}{#2}{\Glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\GLSsymbol}{\@gls@hyp@opt\@GLSsymbol}
\newcommand*{\@GLSsymbol}[2][\%
    \new@ifnextchar[{\@GLSsymbol@{#1}{#2}}{\@GLSsymbol@{#1}{#2}[]}]
\def\@GLSsymbol@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrysymbol{#2}#3}}%
}
\newrobustcmd*{\glssymbolplural}{\@gls@hyp@opt\@glssymbolplural}
\newcommand*{\@glssymbolplural}[2][\%
    \new@ifnextchar[{\@glssymbolplural@{#1}{#2}}{\@glssymbolplural@{#1}{#2}[]}]
\def\@glssymbolplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\Glsymbolplural}{\@gls@hyp@opt\@Glsymbolplural}
\newcommand*{\@Glsymbolplural}[2][\%
    \new@ifnextchar[{\@Glsymbolplural@{#1}{#2}}{\@Glsymbolplural@{#1}{#2}[]}]
\def\@Glsymbolplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
\newcommand*{\@GLSsymbolplural}[2][\%
    \new@ifnextchar[{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2}[]}]
\def\@GLSsymbolplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrysymbolplural{#2}#3}}%
}
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
\newcommand*{\@glsuseri}[2][\%
    \new@ifnextchar[{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2}[]}]
\def\@glsuseri@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
\newcommand*{\@Glsuseri}[2][\%
    \new@ifnextchar[{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2}[]}]
\def\@Glsuseri@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\newcommand*{\@GLSuseri}[2][\%
    \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2}[]}]
\def\@GLSuseri@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseri{#2}#3}}%
}
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
\newcommand*{\@glsuserii}[2][\%
    \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2}[]}]
\def\@glsuserii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
}

```

```

\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
\newcommand*{\@Glsuserii}[2] [] {%
  \new@ifnextchar [{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2} []}]
\def\@Glsuserii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\GLSuserii}{\@gls@hyp@opt\@GLSuserii}
\newcommand*{\@GLSuserii}[2] [] {%
  \new@ifnextchar [{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2} []}]
\def\@GLSuserii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuserii{#2}#3}}%
}
\newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}
\newcommand*{\@glsuseriii}[2] [] {%
  \new@ifnextchar [{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2} []}]
\def\@glsuseriii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
\newcommand*{\@Glsuseriii}[2] [] {%
  \new@ifnextchar [{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2} []}]
\def\@Glsuseriii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\GLSuseriii}{\@gls@hyp@opt\@GLSuseriii}
\newcommand*{\@GLSuseriii}[2] [] {%
  \new@ifnextchar [{\@GLSuseriii@{#1}{#2}}{\@GLSuseriii@{#1}{#2} []}]
\def\@GLSuseriii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuseriii{#2}#3}}%
}
\newrobustcmd*{\glsuseriv}{\@gls@hyp@opt\@glsuseriv}
\newcommand*{\@glsuseriv}[2] [] {%
  \new@ifnextchar [{\@glsuseriv@{#1}{#2}}{\@glsuseriv@{#1}{#2} []}]
\def\@glsuseriv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
\newcommand*{\@Glsuseriv}[2] [] {%
  \new@ifnextchar [{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2} []}]
\def\@Glsuseriv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\GLSuseriv}{\@gls@hyp@opt\@GLSuseriv}
\newcommand*{\@GLSuseriv}[2] [] {%
  \new@ifnextchar [{\@GLSuseriv@{#1}{#2}}{\@GLSuseriv@{#1}{#2} []}]
\def\@GLSuseriv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuseriv{#2}#3}}%
}
\newrobustcmd*{\glsuserv}{\@gls@hyp@opt\@glsuserv}
\newcommand*{\@glsuserv}[2] [] {%

```

```

\new@ifnextchar [{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2} []}]
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\Glsuserv}{\@gls@hyp@opt\@Glsuserv}
\newcommand*{\@Glsuserv}[2] []{%
\new@ifnextchar [{\@Glsuserv@{#1}{#2}}{\@Glsuserv@{#1}{#2} []}]
\def\@Glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserv{#2}#3}%
}
\newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
\newcommand*{\@GLSuserv}[2] []{%
\new@ifnextchar [{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2} []}]
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuserv{#2}#3}}%
}
\newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
\newcommand*{\@glsuservi}[2] []{%
  \new@ifnextchar [{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2} []}]
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
\newrobustcmd*{\Glsuservi}{\@gls@hyp@opt\@Glsuservi}
\newcommand*{\@Glsuservi}[2] []{%
  \new@ifnextchar [{\@Glsuservi@{#1}{#2}}{\@Glsuservi@{#1}{#2} []}]
\def\@Glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuservi{#2}#3}%
}
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\@ns@acrshort}
\newcommand*{\@ns@acrshort}[2] []{%
  \new@ifnextchar [{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2} []}]
}
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@secondoftwo
    \let\gls@scaps\@firstofthree
    \let\glsinsert\@empty
    \def\gls@customtext{%
      \acronymfont{\glsentryshort{#2}#3}%
    }%
    \@gls@link{#1}{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
}

```

```

\glspostlinkhook
}
\newrobustcmd*{\Acrshort}{\@gls@hyp@opt\ns@Acrshort}
\newcommand*{\ns@Acrshort}[2] [] {%
  \new@ifnextchar[{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2} []}]%
}
\def\@Acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsifcaps\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshort}{\@gls@hyp@opt\ns@ACRshort}
\newcommand*{\ns@ACRshort}[2] [] {%
  \new@ifnextchar[{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}]%
}
\def\@ACRshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsifcaps\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\glsentryshort{#2}}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrshortpl}{\@gls@hyp@opt\ns@acrshortpl}
\newcommand*{\ns@acrshortpl}[2] [] {%
  \new@ifnextchar[{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} []}]%
}
\def\@acrshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo

```

```

\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \acronymfont{\glentryshortpl{#2}}#3%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\ns@Acrshortpl}
\newcommand*{\ns@Acrshortpl}[2] [] {%
  \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2} []}%
}
\def\@Acrshortpl#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\ns@ACRshortpl}
\newcommand*{\ns@ACRshortpl}[2] [] {%
  \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2} []}%
}
\def\@ACRshortpl#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\glentryshortpl{#2}}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrlong}{\@gls@hyp@opt\ns@acrlong}
\newcommand*{\ns@acrlong}[2] [] {%
  \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2} []}%
}

```

```

}
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@firstofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \glsentrylong{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls type @entryfmt\endcsname}%
  }%
  \gls post link hook
}
\newrobustcmd*{\Acr long}{\@gls@hyp@opt\@ns@Acr long}
\newcommand*{\ns@Acr long}[2][{}]{%
  \new@if next char[{\@Acr long{#1}{#2}}{\@Acr long{#1}{#2}}[{}]}%
}
\def\@Acr long#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@secondofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \Glsentrylong{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls type @entryfmt\endcsname}%
  }%
  \gls post link hook
}
\newrobustcmd*{\ACR long}{\@gls@hyp@opt\@ns@ACR long}
\newcommand*{\ns@ACR long}[2][{}]{%
  \new@if next char[{\@ACR long{#1}{#2}}{\@ACR long{#1}{#2}}[{}]}%
}
\def\@ACR long#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@thirdofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \mfirstucMakeUppercase{\glsentrylong{#2}#3}%
    }%
  }%
}

```

```

    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\ns@acrlongpl}
\newcommand*{\ns@acrlongpl}[2][]{%
  \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2}[]}%
}
\def\@acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glsescapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\ns@Acrlongpl}
\newcommand*{\ns@Acrlongpl}[2][]{%
  \new@ifnextchar[{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2}[]}%
}
\def\@Acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glsescapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\ns@ACRlongpl}
\newcommand*{\ns@ACRlongpl}[2][]{%
  \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2}[]}%
}
\def\@ACRlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

```

```

\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glsifcaps\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \mfirstucMakeUppercase{\glsentrylongpl{#2}#3}%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newcommand*{\@gls@entry@field}[2]{%
  \csname glo@\glsdetoklabel{#1}@#2\endcsname
}
\newcommand*{\glsletentryfield}[3]{%
  \letcs{#1}{glo@\glsdetoklabel{#2}@#3}%
}
\newcommand*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@#2}%
    \ifdef\@glo@text
    {%
      \xmakefirstuc{\@glo@text}%
    }%
    {%
      ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
        entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
        label and the field name}%
    }%
  }%
  {%
    ???%
  }%
}
\newcommand*{\glsentryname}[1]{\@gls@entry@field{#1}{name}}
\newrobustcmd*{\Glsentryname}[1]{%
  \@Gls@entryname{#1}%
}
\newcommand*{\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
\newcommand*{\@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@name}%
    \expandafter\@gls@getbody\@glo@text{}\@nil
    \expandafter\ifx\@gls@body\glsentrylong\relax
    \expandafter\Glsentrylong\@gls@rest
  }%
  \else

```



```

\expandafter\ifx\@gls@body\glsentryshort\relax
\expandafter\Glsentryshort\@gls@rest
\else
\expandafter\ifx\@gls@body\acronymfont\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\xmakefirstuc{\@glo@text}%
\fi
\fi
\fi
\fi
\fi
}%
{%
\@Gls@entry@field{#1}{name}%
}%
}
\newcommand*\glsentrydesc}[1]{\@Gls@entry@field{#1}{desc}}
\newrobustcmd*\Glsentrydesc}[1]{%
\@Gls@entry@field{#1}{desc}%
}
\newcommand*\glsentrydescplural}[1]{%
\@Gls@entry@field{#1}{descplural}%
}
\newrobustcmd*\Glsentrydescplural}[1]{%
\@Gls@entry@field{#1}{descplural}%
}
\newcommand*\glsentrytext}[1]{\@Gls@entry@field{#1}{text}}
\newrobustcmd*\Glsentrytext}[1]{%
\@Gls@entry@field{#1}{text}%
}
\newcommand*\glsentryplural}[1]{%
\@Gls@entry@field{#1}{plural}%
}
\newrobustcmd*\Glsentryplural}[1]{%
\@Gls@entry@field{#1}{plural}%
}
\newcommand*\glsentrysymbol}[1]{%
\@Gls@entry@field{#1}{symbol}%
}
\newrobustcmd*\Glsentrysymbol}[1]{%
\@Gls@entry@field{#1}{symbol}%
}

```

```

}
\newcommand*\glsentrysymbolplural}[1]{%
  \@gls@entry@field{#1}{symbolplural}%
}
\newrobustcmd*\Glsentrysymbolplural}[1]{%
  \@Gls@entry@field{#1}{symbolplural}%
}
\newcommand*\glsentryfirst}[1]{%
  \@gls@entry@field{#1}{first}%
}
\newrobustcmd*\Glsentryfirst}[1]{%
  \@Gls@entry@field{#1}{first}%
}
\newcommand*\glsentryfirstplural}[1]{%
  \@gls@entry@field{#1}{firstpl}%
}
\newrobustcmd*\Glsentryfirstplural}[1]{%
  \@Gls@entry@field{#1}{firstpl}%
}
\newcommand*\glscapitalisewords}[1]{%
  \@capitalisewords{#1}%
}
\newrobustcmd*\@glsentrytitlecase}[2]{%
  \glsdoifexists{#1}%
  {%
    \glsfieldfetch{#1}{#2}{\@gls@value}%
    \expandafter\glscapitalisewords\expandafter{\@gls@value}%
  }%
}
\ifdef\texorpdfstring
{
  \newcommand*\glsentrytitlecase}[2]{%
    \texorpdfstring
    {\@glsentrytitlecase{#1}{#2}}%
    {\@gls@entry@field{#1}{#2}}%
  }
}
{
  \newcommand*\glsentrytitlecase}[2]{\@glsentrytitlecase{#1}{#2}}
}
\newcommand*\glsentrytype}[1]{\@gls@entry@field{#1}{type}}
\newcommand*\glsentrysort}[1]{%
  \@gls@entry@field{#1}{sort}%
}
\newcommand*\glsentryparent}[1]{%
  \@gls@entry@field{#1}{parent}%
}
\newcommand*\glsentryuseri}[1]{%
  \@gls@entry@field{#1}{useri}%
}
}

```

```

\newrobustcmd*\Glsentryuseri}[1]{%
  \@Gls@entry@field{#1}{useri}%
}
\newcommand*\glsentryuserii}[1]{%
  \@gls@entry@field{#1}{userii}%
}
\newrobustcmd*\Glsentryuserii}[1]{%
  \@Gls@entry@field{#1}{userii}%
}
\newcommand*\glsentryuseriii}[1]{%
  \@gls@entry@field{#1}{useriii}%
}
\newrobustcmd*\Glsentryuseriii}[1]{%
  \@Gls@entry@field{#1}{useriii}%
}
\newcommand*\glsentryuseriv}[1]{%
  \@gls@entry@field{#1}{useriv}%
}
\newrobustcmd*\Glsentryuseriv}[1]{%
  \@Gls@entry@field{#1}{useriv}%
}
\newcommand*\glsentryuserv}[1]{%
  \@gls@entry@field{#1}{userv}%
}
\newrobustcmd*\Glsentryuserv}[1]{%
  \@Gls@entry@field{#1}{userv}%
}
\newcommand*\glsentryuservi}[1]{%
  \@gls@entry@field{#1}{uservi}%
}
\newrobustcmd*\Glsentryuservi}[1]{%
  \@Gls@entry@field{#1}{uservi}%
}
}
\newcommand*\glsentryshort}[1]{\@gls@entry@field{#1}{short}}
\newrobustcmd*\Glsentryshort}[1]{%
  \@Gls@entry@field{#1}{short}%
}
}
\newcommand*\glsentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}
\newrobustcmd*\Glsentryshortpl}[1]{%
  \@Gls@entry@field{#1}{shortpl}%
}
}
\newcommand*\glsentrylong}[1]{\@gls@entry@field{#1}{long}}
\newrobustcmd*\Glsentrylong}[1]{%
  \@Gls@entry@field{#1}{long}%
}
}
\newcommand*\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}
\newrobustcmd*\Glsentrylongpl}[1]{%
  \@Gls@entry@field{#1}{longpl}%
}
}
\newcommand*\glsentryfull}[1]{%

```

```

\acrfullformat{\glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
}
\newrobustcmd*{\Glsentryfull}[1]{%
\acrfullformat{\Glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
}
\newcommand*{\glsentryfullpl}[1]{%
\acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newrobustcmd*{\Glsentryfullpl}[1]{%
\acrfullformat{\Glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newcommand*{\glsentrynumberlist}[1]{%
\glsdoifexists{#1}%
{%
\@gls@entry@field{#1}{numberlist}%
}%
}
\@ifpackageloaded{hyperref} {%
\newcommand*{\glsdisplaynumberlist}[1]{%
\GlossariesWarning
{%
\string\glsdisplaynumberlist\space
doesn't work with hyperref.^^JUsing
\string\glsentrynumberlist\space instead%
}%
\glsentrynumberlist{#1}%
}%
}%
{%
\newcommand*{\glsdisplaynumberlist}[1]{%
\glsdoifexists{#1}%
{%
\bgroup
\protected@edef\@glo@label{\glsdetoklabel{#1}}%
\let\@org@glsnumberformat\glsnumberformat
\def\glsnumberformat##1{##1}%
\protected@edef\the@numberlist{%
\csname glo@\@glo@label @numberlist\endcsname}%
\def\@gls@numlist@sep{}%
\def\@gls@numlist@nextsep{}%
\def\@gls@numlist@lastsep{}%
\def\@gls@thislist{}%
\def\@gls@donext@def{}%
\renewcommand\do[1]{%
\protected@edef\@gls@thislist{%
\@gls@thislist
\noexpand\@gls@numlist@sep
##1%
}%
\let\@gls@numlist@sep\@gls@numlist@nextsep

```

```

        \def\@gls@numlist@nextsep{\glsnumlistsep}%
        \@gls@donext@def
        \def\@gls@donext@def{%
            \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
        }%
    }%
\expandafter \glsnumlistparser \expandafter{\the@numberlist}%
\let\@gls@numlist@sep\@gls@numlist@lastsep
\@gls@thislist
\egroup
}%
}
}
\newcommand*{\glsnumlistsep}{, }
\newcommand*{\glsnumlistlastsep}{ \& }
\newcommand*{\gls hyperlink}[2][\glsentrytext{\@glo@label}]{%
    \def\@glo@label{#2}%
    \@glslink{\glo@linkprefix\glsdetoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*{\glsadd}[2][{%
    \@gls@adjustmode
    \glsdoifexists{#2}%
    {%
        \def\@glsnumberformat{glsnumberformat}%
        \protected@edef\@gls@counter{\csname glo@\glsdetoklabel{#2}@counter\endcsname}%
        \setkeys{glossadd}{#1}%
        \@gls@saveentrycounter
        \@gls@setsort{#2}%
        \@do@wrglossary{#2}%
    }%
}
}
\newcommand*{\@gls@adjustmode}{}
\AtBeginDocument{\renewcommand*{\@gls@adjustmode}{\ifvmode\mbox{}\fi}}
\newrobustcmd*{\glsaddall}[1][{%
    \protected@edef\@glo@type{\@glo@types}%
    \setkeys{glossadd}{#1}%
    \forallglsentries[\@glo@type]{\@glo@entry}{%
        \glsadd[#1]{\@glo@entry}%
    }%
}
}
\newrobustcmd*{\glsaddallunused}[1][\@glo@types]{%
    \forallglsentries[#1]{\@glo@entry}%
    {%
        \ifglsused{\@glo@entry}{\glsadd[format=glsignore]{\@glo@entry}}%
    }%
}
}
\newcommand*{\glsignore}[1]{}
\edef\glsopenbrace{\expandafter\@gobble\string\{}}

```

```

\edef\glsclosebrace{\expandafter@gobble\string\}}
\edef\glsbackslash{\expandafter@gobble\string\}
\edef\glsquote#1{\string"#1\string"}
\edef\glspercentchar{\expandafter@gobble\string\%}
\edef\glsstildechar{\string~}
\ifglsexindy
  \newcommand*{\@glsfirstletter}{A}
\fi
\newcommand*{\GlsSetXdyFirstLetterAfterDigits}{%
  \@ifstar\s@GlsSetXdyFirstLetterAfterDigits@GlsSetXdyFirstLetterAfterDigits}
\ifglsexindy
  \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*{\@glsfirstletter}{#1}}
  \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*{\@glsfirstletter}{#1}%
    \@onelevel@sanitize\@glsfirstletter
  }
\else
  \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \glsnoindywarning\GlsSetXdyFirstLetterAfterDigits}
  \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%
    \@GlsSetXdyFirstLetterAfterDigits
  }
\fi
\ifglsexindy
  \newcommand*{\@xdynumbergrouporder}{:before \string"@glsfirstletter\string"}
\fi
\newcommand*{\GlsSetXdyNumberGroupOrder}[1]{%
  \@ifstar\s@GlsSetXdyNumberGroupOrder@GlsSetXdyNumberGroupOrder
}
\ifglsexindy
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
  }
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
    \@onelevel@sanitize\@xdynumbergrouporder
  }
\else
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \glsnoindywarning\GlsSetXdyNumberGroupOrder}
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
    \@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*{\@glsminrange}{2}
\ifglsexindy
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \renewcommand*{\@glsminrange}{#1}}
\else
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%

```



```

        ^^J\space\space\space
        :min-range-length \@glsminrange^^J%
    )
    }%
} %
} %
} %
} %
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
:unverified )}%
\write\glswrite{(markup-crossref-list
: class \string"see\string"^^J\space\space\space
: open \string"\string\glsseeformat\string"
: close \string"{}\string")}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
(\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
: open \string"\string
\glossarysection[\string\glossarytoctitle]{\string
\glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
{%
\@for\@this@attr:=\@xdyattributelist\do{%
\protected\write\glswrite{\string\providecommand*%
\expandafter\string
\csname glsX\@this@ctr X\@this@attr\endcsname[2]%
{%
\string\setentrycounter
[\expandafter\@gobble\string\#1]{\@this@ctr}%
\expandafter\string
\csname\@this@attr\endcsname
{\expandafter\@gobble\string\#2}%
}%
}%
}%
}%
\write\glswrite{%
\string\begin
{theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
\space\space:close \string"\glspercentchar\glstildechar n\string
\end{theglossary}\string\glossarypostamble
\glstildechar n\string" ^^J\space\space\space
:tree}%
\write\glswrite{(markup-letter-group-list

```



```

        :sep \string"\string\glsgroupskip\glstildechar n\string"}}%
\write\glswrite{(markup-indexentry
        :open \string"\string\relax \string\glsresetentrylist
        \glstildechar n\string"}}%
\write\glswrite{(markup-locclass-list :open
        \string"\glsopenbrace\string\glossaryentrynumbers
        \glsopenbrace\string\relax\space \string"^^J\space\space\space
        :sep \string", \string"
        :close \string"\glsclosebrace\glsclosebrace\string"}}%
\write\glswrite{(markup-locref-list
        :sep \string"\string\delimN\space\string"}}%
\write\glswrite{(markup-range
        :sep \string"\string\delimR\space\string"}}%
\@onelevel@sanitize\gls@suffiF
\@onelevel@sanitize\gls@suffiFF
\ifx\gls@suffiF\@empty
\else
        \write\glswrite{(markup-range
                :close "\gls@suffiF" :length 1 :ignore-end)}}%
\fi
\ifx\gls@suffiFF\@empty
\else
        \write\glswrite{(markup-range
                :close "\gls@suffiFF" :length 2 :ignore-end)}}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
        :sep \string"\string\glsgroupskip\glstildechar n\string"}}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
        :open-head \string"\string\glsgroupheading
        \glsopenbrace\string"^^J\space\space\space
        :close-head \string"\glsclosebrace\string"}}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}%
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax

```

```

\ifundef{\glswrite}{\newwrite\glswrite}{\relax
\openout\glswrite=\istfilename
\write\glswrite{\glspercentchar\space makeindex style file
  created by the glossaries package}
\write\glswrite{\glspercentchar\space for document
  'jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '@gls@actualchar'}
\write\glswrite{encap '@gls@encapchar'}
\write\glswrite{level '@gls@levelchar'}
\write\glswrite{quote '@gls@quotechar'}
\write\glswrite{keyword \string\string\glossaryentry\string}
\write\glswrite{preamble \string\string\glossarysection[\string
  \glossarytoctitle]{\string\glossarytitle}\string
  \glossarypreamble\string\n\string\begin{theglossary}\string
  \glossaryheader\string\n\string}
\write\glswrite{postamble \string\string%\string\n\string
  \end{theglossary}\string\glossarypostamble\string\n
  \string}
\write\glswrite{group_skip \string\string\glsgroupskip\string\n
  \string}
\write\glswrite{item_0 \string\string%\string\n\string}
\write\glswrite{item_1 \string\string%\string\n\string}
\write\glswrite{item_2 \string\string%\string\n\string}
\write\glswrite{item_01 \string\string%\string\n\string}
\write\glswrite{item_x1
  \string\string\relax \string\glsresetentrylist\string\n
  \string}
\write\glswrite{item_12 \string\string%\string\n\string}
\write\glswrite{item_x2
  \string\string\relax \string\glsresetentrylist\string\n
  \string}
\write\glswrite{delim_0 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_1 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_2 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_t \string\string\}\string}\string}
\write\glswrite{delim_n \string\string\delimN \string}
\write\glswrite{delim_r \string\string\delimR \string}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string\string\glsgroupheading\string\}\string}
\write\glswrite{heading_suffix
  \string\string}\string\relax
  \string\glsresetentrylist \string}
\write\glswrite{symhead_positive \string\glsymbols\string}
\write\glswrite{numhead_positive \string\glsnumbers\string}
\write\glswrite{page_compositor \string\glscompositor\string}
\@gls@escsdq\gls@suffixF

```

```

\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\fi
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{-}
\ifglsxindy
\newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\newcommand*\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}%
\ifpackageloaded{tracklang}%
{%
  \IfTrackedLanguage{german}%
  {%
    \def\@gls@extramakeindexopts{-g}%
  }%
  {}%
}%
{}%
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkquote####1#####2#1#####3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={####1}%
\noexpand\ifx\noexpand\null####2\noexpand\null
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%

```

```

\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###2#1#1\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###2#1###3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\renewcommand\noexpand\@gls@checkmkidxchars}[1]{%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkquote###1\noexpand\@nil
#1#1\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkesquote###1\noexpand\@nil
\expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkesactual###1\noexpand\@nil
\noexpand\?\noexpand\?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkactual###1\noexpand\@nil
\noexpand?\noexpand?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkbar###1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkeschar###1\noexpand\@nil
\noexpand\|\noexpand\|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checklevel###1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null

```

```

\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkesclevel###1\noexpand\@nil
\noexpand\!\noexpand\!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkesc###1\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkescquote###1%
\expandonce{\csname#1\endcsname}###2\expandonce{\csname#1\endcsname}%
###3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={###1}%
\noexpand\ifx\noexpand\null###2\noexpand\null
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand|string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand|string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand|string
\expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###2\expandonce{\csname#1\endcsname}%
\expandonce{\csname#1\endcsname}\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###2\expandonce{\csname#1\endcsname}%
###3\noexpand\null}%
\noexpand\fi

```

```

        \noexpand\fi
        \noexpand\@@gls@checkescquote
    }%
}
\@gls@docheckquotedef
}
\newcommand*\@gls@nosetquote}[1]{\PackageError{glossaries}%
    {\string\GlsSetQuote\space not permitted here}%
    {Move \string\GlsSetQuote\space earlier in the preamble, as
    soon as possible after glossaries.sty has been loaded}}
\fi
\newcommand*\@gls@extramakeindexopts}[1]{}
\newcommand\@noist{%
    \@gls@addpredefinedattributes
    \let\writeist\relax
}
\newcommand*\@makeglossary}[1]{%
    \ifglossaryexists{#1}%
    {%
        \ifglssavewrites
            \expandafter\newtoks\csname glo@#1@filetok\endcsname
        \else
            \expandafter\newwrite\csname glo@#1@file\endcsname
            \expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
        \fi
        \@gls@renewglossary
        \writeist
    }%
    {%
        \PackageError{glossaries}%
        {Glossary type ‘#1’ not defined}%
        {New glossaries must be defined before using \string\makeglossaries}%
    }%
}
\newcommand*\@glsopenfile}[2]{%
    \immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
    \PackageInfo{glossaries}{Writing glossary file
        \jobname.\csname @glotype@#2@out\endcsname}%
}
\newcommand*\@closegls}[1]{%
    \closeout\csname glo@#1@file\endcsname
}
\ifglxindy
\newcommand*\@gls@automake}[1]{%
    \ifglossaryexists{#1}
    {%
        \@closegls{#1}%
        \ifdefstring{\glsorder}{letter}%
        {\def\@gls@order{-M ord/letorder }}%
        {\let\@gls@order\@empty}%
    }
}

```

```

\ifcsundef{@xdy@#1@language}%
  {\let\@gls@langmod\@xdy@main@language}%
  {\letcs\@gls@langmod{@xdy@#1@language}}%
\edef\@gls@dothiswrite{\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \@gls@istfilebase\space
-C \@gls@codepage\space
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}%
}%
\@gls@dothiswrite
}%
{
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\else
\newcommand*{\@gls@automake}[1]{%
\ifglossaryexists{#1}
{%
\@closegls{#1}%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-l }}%
{\let\@gls@order\@empty}%
\edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
-s \istfilename\space
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}%
}%
\@gls@dothiswrite
}%
{
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\fi
\ifglsxindy
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%
\IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-M ord/letorder }}%
{\let\@gls@order\@empty}%
\ifcsundef{@xdy@#1@language}%

```

```

        {\let\@gls@langmod\@xdy@main@language}%
        {\letcs\@gls@langmod{\@xdy@#1@language}}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \gls@istfilebase\space
-C \gls@codepage\space
-t \jobname.\csuse{@gloftype@#1@log}
-o \jobname.\csuse{@gloftype@#1@in}
\jobname.\csuse{@gloftype@#1@out}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@gloftype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%
\IfFileExists{\jobname.\csuse{@gloftype@#1@out}}%
{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-l }}%
{\let\@gls@order\@empty}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order
-s \istfilename\space
-t \jobname.\csuse{@gloftype@#1@log}
-o \jobname.\csuse{@gloftype@#1@in}
\jobname.\csuse{@gloftype@#1@out}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@gloftype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\fi
\newcommand*{\@warn@nomakeglossaries}{}
\newcommand*{\warn@nomakeglossaries}{\@warn@nomakeglossaries}
\newcommand{\@gls@automake@immediate}{%
\ifnum\gls@automake@nr=2\relax

```



```

\@for\@gls@type:=\@glo@types\do{%
  \ifdefempty{\@gls@type}{}%
  {\@gls@automake@immediate{\@gls@type}}%
}%
\glsautomakefalse
\renewcommand*{\@gls@doautomake}{}%
\fi
}
\newcommand*{\makeglossaries}{%
\@domakeglossaries
{%
  \@gls@@automake@immediate
  \ifundef{\glswrite}{\newwrite\glswrite}{}%
  \protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{}}
  \protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{}}
  \ifundef{\@gls@extramakeindexopts
  }{%
  {%
    \protected@write\@auxout{}{\string\providecommand
      \string\@gls@extramakeindexopts[1]{}}
    \protected@write\@auxout{}{\string\@gls@extramakeindexopts
      {\@gls@extramakeindexopts}}%
  }%
  \protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
  \protected@write\@auxout{}{\string\@glsorder{\glsorder}}
  \@for\@glo@type:=\@glo@types\do{%
    \ifthenelse{\equal{\@glo@type}{}}{ }{%
      \@makeglossary{\@glo@type}}%
    }%
  \renewcommand*\newglossary[4] [] {%
  \PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%
\let\@makeglossary\@gobble
\let\makeglossaries\relax
\@disable@onlypremakeg
\let\gls@checkseeallowed\relax
\let\warn@nomakeglossaries\relax
\let\warn@noprntglossary\@makeglossaries@warn@noprntglossary
\ifglssavenumberlist
  \edef\@gls@dodolistparser{\noexpand\DeclareListParser
    {\noexpand\glsnumlistparser}{\delimN}}%
  \@gls@dodolistparser
\fi
\let\makenoidxglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
  \let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
\ifglsautomake

```

```

\renewcommand*{\@gls@doautomake}{%
  \for\@gls@type:=\@glo@types\do{%
    \ifdefempty{\@gls@type}{}%
    {\@gls@automake{\@gls@type}}%
  }%
}%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}
\newcommand\@makeglossaries@warn@noprintglossary{%
  \ifdefstring{\@glo@types}{,}%
  {%
    \GlossariesWarningNoLine{No glossaries have been defined}%
  }%
  {%
    \GlossariesWarningNoLine{No \string\printglossary\space
      or \string\printglossaries\space
      found. ^^J(Remove \string\makeglossaries\space if you
      don't want any glossaries.) ^^JThis document will not
      have a glossary}%
  }%
}%
\AtEndDocument{%
  \warn@nomakeglossaries
  \warn@noprintglossary
}
\newcommand*{\makenoidxglossaries}{%
  \@domakeglossaries
  {%
    \renewcommand{\@gls@noref@warn}[1]{%
      \GlossariesWarning{Empty glossary for
        \string\printnoidxglossary[type={##1}].
        Rerun may be required (or you may have forgotten to use
        commands like \string\gls)}%
    }%
    \let\@gls@checkmkidxchars\@gobble
    \glsesclocationsfalse
    \let\@do@@wrglossary\gls@noidxglossary
    \let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
    \let\gls@checkseeallowed\relax
    \renewcommand{\@do@seeglossary}[2]{%
      \protected@edef\@gls@label{\glsdetoklabel{##1}}%
      \protected@write\@auxout{}{%
        \string\@gls@reference
        {\csname glo@\@gls@label @type\endcsname}%
        {\@gls@label}}%
      {%
        \string\glsseeformat##2}%
    }%
  }%
}

```

```

    }%
  }%
}%
\AtBeginDocument
{%
  \write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
}%
\def\warn@noprintglossary{%
  \GlossariesWarningNoLine{No \string\printnoidxglossary\space
    or \string\printnoidxglossaries ^^J
    found. (Remove \string\makenoidxglossaries\space if you
    don't want any glossaries.)^^JThis document will not have a glossary}%
}%
\let\warn@nomakeglossaries\relax
\let\makeglossaries\@no@makeglossaries
\renewcommand*\@printgloss@setsort{%
  \let\@glo@assign@sortkey\@glo@assign@sortkey
  \def\@glo@sorttype{\@glo@default@sorttype}%
}%
\renewcommand*\new@glossaryentry[2]{%
  \PackageError{glossaries}{Glossary entries must be
    defined in the preamble^^Jwhen you use
    \string\makenoidxglossaries}%
  {Either move your definitions to the preamble or use
    \string\makeglossaries}%
}%
\renewcommand*\@gls@entrynumberlist[1]{%
  \letcs{\@gls@loclist}{glo@\@gls@detoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \glsnoidxloclist{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for '##1'. Either
        a rerun is required or you haven't referenced the entry}%
    }%
  }%
}%
\renewcommand*\@gls@displaynumberlist[1]{%
  \letcs{\@gls@loclist}{glo@\@gls@detoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \def\@gls@noidxloclist@sep{%
      \def\@gls@noidxloclist@sep{%
        \def\@gls@noidxloclist@sep{%
          \glsnumlistsep
        }%
        \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
      }%
    }%
  }%

```

```

    }%
  }%
  \def\@gls@noidxloclist@finalsep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
  \@gls@noidxloclist@finalsep
  \@gls@noidxloclist@prev
}%
{
  ??\glsdoifexists{##1}%
  {
    \GlossariesWarning{Missing location list for ‘##1’. Either
      a rerun is required or you haven’t referenced the entry}%
  }%
}%
}%
\renewcommand*\@glsnumberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo\@glsdetoklabel{##1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
  \let\@gls@org@glsseeformat\glsseeformat
  \let\glsnoidxdisplayloc##2\relax
  \let\glsseeformat##3\relax
  \ifdef\@gls@loclist
  {
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {
    ??\glsdoifexists{##1}%
    {
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
  \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
  \let\glsseeformat\@gls@org@glsseeformat
}%
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
\@glo@check@sortallowed\makenoidxglossaries
}%
}
\@onlypreamble{\makenoidxglossaries}
\newcommand*\@glsnumberlistloop}[2]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
\newcommand*\@glsnoidxnumberlistloophandler}[1]{%
  #1%
}

```

```

\newcommand*\@no@makeglossaries}{%
  \PackageError{glossaries}{You can't use both
  \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
  together.}%
}
\newcommand{\@gls@nofwarn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
  is required to make \string\printnoidxglossary[type={#1}] work}%
}
\newcommand*\@gls@noidxglossary}{%
  \protected@write\@auxout{%
    \string\@gls@reference
    {\csname glo@\@gls@label @type\endcsname}%
    {\@gls@label}%
    {\string\glsnoidxdisplayloc
    {\@gls@counterprefix}%
    {\@gls@counter}%
    {\@glsnumberformat}%
    {\@glslocref}}%
  }%
}
}
\providecommand\istfile{\glswrite}
\AtEndDocument{%
  \glswritefiles
}
\newcommand*\@gls@writefiles{%
  \forallglossaries{\@glo@type}{%
    \ifcsundef{glo@\@glo@type @filetok}%
    {%
      \def\gls@tmp{}%
    }%
    {%
      \edef\gls@tmp{\expandafter\the
        \csname glo@\@glo@type @filetok\endcsname}%
    }%
    \ifx\gls@tmp\@empty
      \ifx\@glo@type\glsdefaulttype
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries.^^JRemember to use package option 'nomain' if
you
          don't want to^^Juse the main glossary}%
      \else
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries}%
      \fi
    \else
      \@glsopenfile{\glswrite}{\@glo@type}%
      \immediate\write\glswrite{%

```

```

        \expandafter\the
        \csname glo@\@glo@type @filetok\endcsname}%
    \immediate\closeout\glswrite
\fi
}%
}
\newcommand*\@gls@glossary}[1]{%
    \@gls@glossary{#1}%
}
\newcommand*\@gls@glossary}[2]{%
    \if@gls@debug
        \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
    \fi
}
\newcommand{\@gls@renewglossary}{%
    \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
    \let\@gls@renewglossary\@empty
}
\newcommand*\@gls@wrglossary}[2]{%
    \ifgls@savewrites
        \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
        \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
            \expandafter{\@gls@tmp^^J}%
    \else
        \ifcsdef{glo@#1@file}%
            {%
                \expandafter\protected@write\csname glo@#1@file\endcsname{%
                    \gls@disablepagerefexpansion}{#2}%
            }%
            {%
                \ifignoredglossary{#1}{}%
                    {%
                        \GlossariesWarning{No file defined for glossary ‘#1’}%
                    }%
            }%
        \fi
    \endgroup\@esphack
}
\newcommand*\@do@wrglossary}[1]{%
    \glswriteentry{#1}{\@do@wrglossary{#1}}%
}
\newcommand*\@gls@writeentry}[2]{%
    \ifgls@indexonlyfirst
        \ifgls@used{#1}{#2}%
    \else
        #2%
    \fi
}
\newcommand{\@gls@protected@pagefmts}{\@gls@numberpage,\@gls@alphpage,%
\@gls@Alphpage,\@gls@romanpage,\@gls@Romanpage,\@gls@arabicpage}

```

```

\newcommand*\gls@disablepagerefexpansion}{%
  \for\@gls@this:=\gls@protected@pagefmts\do
  {%
    \expandafter\let\@gls@this\relax
  }%
}
\newcommand*\gls@alphpage}{\@alph\c@page}
\newcommand*\gls@Alphpage}{\@Alph\c@page}
\newcommand*\gls@numberpage}{\number\c@page}
\newcommand*\gls@arabicpage}{\@arabic\c@page}
\newcommand*\gls@romanpage}{\romannumeral\c@page}
\newcommand*\gls@Romanpage}{\@Roman\c@page}
\newcommand*\gls@saddprotectedpagefmt}[1]{%
  \protected@eappto\gls@protected@pagefmts{\expandonce{\csname gls#1page\endcsname}}%
  \csedef\gls#1page{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
  \protected@eappto\@wrglossarynumberhook{%
    \noexpand\let\expandonce{\csname org@gls#1\endcsname}%
    \expandonce{\csname#1\endcsname}%
    \noexpand\def\expandonce{\csname#1\endcsname}{%
      \noexpand\@wrglossary@pageformat
      \expandonce{\csname gls#1page\endcsname}%
      \expandonce{\csname org@gls#1\endcsname}%
    }%
  }%
}
\newcommand*\@wrglossarynumberhook{}
\newcommand{\@wrglossary@pageformat}[3]{%
  \ifx#3\c@page #1\else #2#3\fi
}
\newcommand*\@do@wrglossary}[1]{%
  \ifglsclocations
    \@do@esc@wrglossary{#1}%
  \else
    \@do@noesc@wrglossary{#1}%
  \fi
}
\newcommand*\@do@noesc@wrglossary}[1]{%
  \expandafter\def\expandafter\@glslocref\expandafter{\theglsentrycounter}%
  \expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
  \ifx\@glsHlocref\@glslocref
    \def\@glo@counterprefix{}%
  \else
    \protected@edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
      {\@glslocref}{\@glsHlocref}%
    }%
    \@do@gls@getcounterprefix
  \fi
  \protected@edef\@gls@label{\glsdetoklabel{#1}}%
  \@do@wrglossary
}

```

```

\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodstrue
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
\begingroup
\let\gls@orgthe\the
\let\gls@orgnumber\number
\let\gls@orgarabic\@arabic
\let\gls@orgromannumeral\romannumeral
\let\gls@orgalph\@alph
\let\gls@orgAlph\@Alph
\let\gls@orgRoman\@Roman
\ifglswrallowprimitivemods
\def\gls@the##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
\def\the{\expandafter\gls@the}%
\def\gls@number##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
\def\number{\expandafter\gls@number}%
\fi
\def\@arabic##1{%
\ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\romannumeral##1{%
\ifx##1\c@page \gls@romanpage\else\gls@orgromannumeral##1\fi}%
\def\@Roman##1{%
\ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
\ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
\ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\@glslocref{\theHglstentrycounter}%
\endgroup
\@gls@checkmkidxchars\@glslocref
\expandafter\ifx\theHglstentrycounter\theHglstentrycounter\relax
\def\@glo@counterprefix{}%
\else
\protected@edef\@glsHlocref{\theHglstentrycounter}%
\@gls@checkmkidxchars\@glsHlocref
\edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
{\@glslocref}{\@glsHlocref}%
}%
\@do@gls@getcounterprefix
\fi
\protected@edef\@gls@label{\glsdetoklabel{#1}}%
\@do@wrglossary
}
\newcommand*{\@do@wrglossary}{%
\ifglswrallowprimitivemods
\expandafter\@glo@checkmkidxrangechar\@glsnumberformat\@nil

```



```

\def\@glo@range{}%
\expandafter\if\@glo@prefix(\relax
\def\@glo@range{:open-range}%
\else
\expandafter\if\@glo@prefix)\relax
\def\@glo@range{:close-range}%
\fi
\fi
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
(indexentry :tkey (\csname glo@\@gls@label @index\endcsname)
:locref \string"\@glo@counterprefix}{\@gls@locref}\string" %
:attr \string"\@gls@counter\@glo@suffix\string"
\@glo@range
)
}%
\else
\@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@gls@numberformat}%
{\@glo@counterprefix}%
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
\string@glossaryentry{\csname glo@\@gls@label @index\endcsname
\@gls@encapchar\@glo@numfmt}{\@gls@locref}}%
\fi
}
\newcommand*\@gls@getcounterprefix[2]{%
\edef\@gls@thisloc{#1}\edef\@gls@thisHloc{#2}%
\ifx\@gls@thisloc\@gls@thisHloc
\def\@glo@counterprefix{}%
\else
\def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
\def\@glo@tmp{##2}%
\ifx\@glo@tmp\@empty
\def\@glo@counterprefix{}%
\else
\def\@glo@counterprefix{##1}%
\fi
}%
\@gls@get@counterprefix#2.#1\end@getprefix
\ifx\@glo@counterprefix\@empty
\GlossariesWarning{Hyper target ‘#2’ can’t be formed by
prefixing~Jlocation ‘#1’. You need to modify the
definition of \string\theH\@gls@counter~Jotherwise you
will get the warning: “name{\@gls@counter.#1}’ has been~J
referenced but does not exist”}%
\fi
\fi
}
\newcommand{\@do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref

```

```

\ifglxindy
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    (indexentry
      :tkey (\csname glo@#1@index\endcsname)
      :xref (\string"\@gls@xref\string")
      :attr \string"see\string"
    )
  }%
\else
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    \string\glossaryentry{\csname glo@#1@index\endcsname
    \@gls@encapchar glsseeformat\@gls@xref}{Z}}%
\fi
}
\def\@gls@fixbraces#1#2#3\@nil{%
  \ifx#2[\relax
    \@gls@fixbraces#1#2#3\@end@fixbraces
  \else
    \def#1{{#2#3}}%
  \fi
}
\def\@gls@fixbraces#1[#2]#3\@end@fixbraces{%
  \def#1{[#2]{#3}}%
}
\newrobustcmd*\glssee}[3][\seename]{%
  \@do@seeglossary{#2}{[#1]{#3}}
\newcommand*\@glssee}[3][\seename]{%
  \glssee[#1]{#3}{#2}}
\newrobustcmd*\glsseeformat}[3][\seename]{%
  \emph{#1} \glsseelist{#2}}
\newrobustcmd*\glsseelist}[1]{%
  \let\@gls@dolast\relax
  \let\@gls@donext\relax
  \@for\@gls@thislabel:=#1\do{%
    \ifx\@xfor@nextelement\@nnil
      \@gls@dolast
    \else
      \@gls@donext
    \fi
    \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
    \let\@gls@dolast\glsseelastsep
    \let\@gls@donext\glsseesep
  }%
}
\newcommand*\glsseelastsep}{\space\andname\space}
\newcommand*\glsseesep}{, }
\newrobustcmd*\glsseeitem}[1]{\glshyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*\glsseeitemformat}[1]{\glsentrytext{#1}}
\newcommand*\gls@save@numberlist}[1]{%
  \ifglssavenumberlist

```

```

\toks@{#1}%
\protected@edef\do@writeaux@info{%
  \noexpand\csgdef{glo@glscurrententrylabel @numberlist}{\the\toks@}%
}%
\@onelevel@sanitize\do@writeaux@info
\protected@write\auxout{}{\do@writeaux@info}%
\fi
}
\newcommand*\warn@noprntglossary{}{}%
\ifcsundef{printglossary}{}{}%
{%
  \@gls@warnonglossdefined
  \undef\printglossary
}
\newcommand*\@printgloss@checkexists@noignored}[2]{%
  \ifglossaryexists{#1}%
  {#2}%
  {%
    \ifignoredglossary{#1}%
    {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
    {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
  }%
}
\newcommand*\@printgloss@checkexists@allowignored}[2]{%
  \s@ifglossaryexists{#1}%
  {#2}%
  {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\newcommand*\printglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@glossary}%
}
\newcommand*\printglossaries{%
  \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}
\newcommand*\printnoidxglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@noidx@glossary}%
}
\newcommand*\printnoidxglossaries{%
  \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
\newcommand*\@printgloss@setsort{}{}
\newcommand*\@gls@preglossaryhook{}{}
\newcommand*\printglossary}[2]{%
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{\csname @glo@type@\@glo@type @title\endcsname}%
  \def\glossarytoctitle{\glossarytitle}%
  \let\org@glossarytitle\glossarytitle

```

```

\def@glossarystyle{%
  \ifx@glossary@default@style\relax
    \GlossariesWarning{No default glossary style provided \MessageBreak
      for the glossary '@glo@type'. \MessageBreak
      Using deprecated fallback. \MessageBreak
      To fix this set the style with \MessageBreak
      \string\setglossarystyle\space or use the \MessageBreak
      style key=value option}%
  \fi
}%
\def@gls@dotoc@title{\gls@set@toc@title{\@glo@type}}%
\let@org@glossary@entry@numbers@glossary@entry@numbers
\bgroup
  \@print@gloss@set@sort
  \setkeys{print@gloss}{#1}%
  \@print@gloss@check@exists{\@glo@type}%
  {%
    \ifx@glossary@title@org@glossary@title
    \else
      \expandafter\let\csname @glo@type@\@glo@type @title\endcsname
        \glossary@title
    \fi
    \let\current@glossary@\@glo@type
    \let@org@glossary@entry@numbers@glossary@entry@numbers
    \let@gls@non@next@pages@gls@non@next@pages
    \let@gls@next@pages@gls@next@pages
    \let\nopost@desc@\nopost@desc
    \gls@dotoc@title
    \@glossary@style
    \let@gls@org@glossary@entry@field@gloss@entry
    \let@gls@org@glossary@sub@entry@field@sub@gloss@entry
    \renewcommand{\gloss@entry}[1]{%
      \protected@xdef\gls@current@entry@label{\gls@detok@label{##1}}%
      \gls@org@glossary@entry@field{##1}%
    }%
    \renewcommand{\sub@gloss@entry}[2]{%
      \protected@xdef\gls@current@entry@label{\gls@detok@label{##2}}%
      \gls@org@glossary@sub@entry@field{##1}{##2}%
    }%
    \@gls@preglossary@hook
    #2%
  }%
\egroup
\global\let@glossary@entry@numbers@org@glossary@entry@numbers
\global\let\warn@no@print@glossary\relax
}
\newcommand{\@print@glossary}{%
  \makeatletter
  \@input@{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%
  \IfFileExists{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%

```

```

}%
{\null}%
\ifglxindy
  \ifcsundef{@xdy@\@glo@type @language}%
  {%
    \edef\@do@auxoutstuff{%
      \noexpand\AtEndDocument{%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\providecommand\string\@xdylanguage[2]{}%
        }%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\@xdylanguage{\@glo@type}{\@xdy@main@language}}%
        }%
      }%
    }%
  }%
  {%
    \edef\@do@auxoutstuff{%
      \noexpand\AtEndDocument{%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\providecommand\string\@xdylanguage[2]{}%
        }%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\@xdylanguage{\@glo@type}{\csname @xdy@\@glo@type
            @language\endcsname}}%
        }%
      }%
    }%
  }%
  \@do@auxoutstuff
  \edef\@do@auxoutstuff{%
    \noexpand\AtEndDocument{%
      \noexpand\immediate\noexpand\write\@auxout{%
        \string\providecommand\string\@gls@codepage[2]{}%
      }%
      \noexpand\immediate\noexpand\write\@auxout{%
        \string\@gls@codepage{\@glo@type}{\@gls@codepage}}%
      }%
    }%
  }%
  \@do@auxoutstuff
\fi
\renewcommand*{\@warn@nomakeglossaries}{%
  \GlossariesWarningNoLine{\string\makeglossaries\space
    hasn't been used,^^Jthe glossaries will not be updated}%
}%
}
\newcommand*{\@glo@sortentries}[2]{%
  \glosortentrieswarning
  \def\@glo@sortinglist{}%
  \def\@glo@sortinghandler{#1}%
  \protected@edef\@glo@type{#2}%
  \forlistcsloop{\@glo@do@sortentries}{\@glsref@#2}%
  \csdef{\@glsref@#2}{%
    \for\@this@label:=\@glo@sortinglist\do{%
      \xifinlistcs{\@this@label}{\@glsref@#2}%
    }%
  }%
}

```

```

    {}%
    {%
      \listcsxadd{@glsref@#2}{\@this@label}%
    }%
    \ifcsdef{@glo@sortingchildren@\@this@label}%
    {%
      \glo@addchildren{#2}{\@this@label}%
    }%
    {}%
  }%
}
\newcommand*{\@glo@addchildren}[2]{%
  \bgroup
  \letcs{\@glo@childlist}{@glo@sortingchildren@#2}%
  \@for\@this@childlabel:=\@glo@childlist\do
  {%
    \xifinlistcs{\@this@childlabel}{@glsref@#1}%
    {}%
    {%
      \listcsxadd{@glsref@#1}{\@this@childlabel}%
    }%
    \ifcsdef{@glo@sortingchildren@\@this@childlabel}%
    {%
      \glo@addchildren{#1}{\@this@childlabel}%
    }%
    {}%
  }%
}
\egroup
}
\newcommand*{\@glo@do@sortentries}[1]{%
  \ifglshasparent{#1}%
  {%
    \protected@edef\@glo@parent{\csuse{glo@glsdetoklabel{#1}@parent}}%
    \ifcsundef{@glo@sortingchildren@\@glo@parent}%
    {%
      \csdef{@glo@sortingchildren@\@glo@parent}{}%
    }%
    {}%
    \expandafter\@glo@sortedinsert
      \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
    \xifinlistcs{\@glo@parent}{@glsref@\@glo@type}%
    {}%
    {}%
    {}%
    \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
  }%
}
{}%
\@glo@sortedinsert{\@glo@sortinglist}{#1}%

```

```

}%
}
\newcommand*{\@glo@sortedinsert}[2]{%
  \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
}%
\newcommand*{\@glo@sorthandler@word}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@letter}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@case}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@nocase}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlicompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sortmacro@word}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@word}{#1}%
  }%
}

```

```

    {%
      \PackageError{glossaries}{Conflicting sort options:^^J
        \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
        \string\printnoidxglossary[sort=word]}{}}%
    }%
  }
\newcommand*{\@glo@sortmacro@letter}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=letter]}{}}%
  }%
}
\newcommand*{\@glo@sortmacro@standard}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \ifcsdef{\@glo@sorthandler@\glsorder}%
    {%
      \@glo@sortentries{\csuse{\@glo@sorthandler@\glsorder}}{#1}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\glsorder’}{}}%
    }%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=standard]}{}}%
  }%
}
\newcommand*{\@glo@sortmacro@case}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@case}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=case]}{}}%
  }%
}
\newcommand*{\@glo@sortmacro@nocase}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
  }%
}

```



```

    {%
      \PackageError{glossaries}{Conflicting sort options:^^J
        \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
        \string\printnoidxglossary[sort=nocase]}{}%
    }%
  }
\newcommand*{\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{}%
  \forglseentries[#1]{\@gls@thislabel}%
  {%
    \xifinlistcs{\@gls@thislabel}{\@glsref@#1}%
    {%
      \listcsadd{\@glo@sortinglist}{\@gls@thislabel}%
    }%
    {%
      }%
    }%
  }%
  \cslet{\@glsref@#1}{\@glo@sortinglist}%
}
\newcommand*{\@glo@sortmacro@def@do}[1]{%
  \ifinlistcs{#1}{\@glsref@\@glo@type}%
  {}%
  {%
    \listcsadd{\@glsref@\@glo@type}{#1}%
  }%
  \ifcsdef{\@glo@sortingchildren@#1}%
  {%
    \@glo@addchildren{\@glo@type}{#1}%
  }%
  {}%
}
\newcommand*{\@glo@sortmacro@use}[1]{%
\newcommand*{\@print@noidx@glossary}{%
  \ifcsdef{\@glsref@\@glo@type}%
  {%
    \ifcsdef{\@glo@sortmacro@\@glo@sorttype}%
    {%
      \csuse{\@glo@sortmacro@\@glo@sorttype}{\@glo@type}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}%
    }%
    \glossarysection[\glossarytoctitle]{\glossarytitle}%
    \glossarypreamble
    \def\@gls@currentlettergroup{}%
    \begin{theglossary}%
    \glossaryheader
    \glsresetentrylist
    \forlistcsloop{\@gls@noidx@do}{\@glsref@\@glo@type}%
    \end{theglossary}%
  }%
}

```

```

    \glossarypostamble
  }%
  {%
    \@gls@noref@warn{\@glo@type}%
  }%
}
\def\glo@grabfirst#1#2\@nil{%
  \def\@gls@firsttok{#1}%
  \ifdefempty\@gls@firsttok
  {%
    \def\@glo@thislettergrp{0}%
  }%
  {%
    \@onelevel@sanitize\@gls@firsttok
    \expandafter\@glo@grabfirst\@gls@firsttok{}}\@nil
  }%
}
\def\@glo@grabfirst#1#2\@nil{%
  \ifdefempty\@glo@thislettergrp
  {%
    \def\@glo@thislettergrp{glssymbols}%
  }%
  {%
    \count@=\ucode‘#1\relax
    \ifnum\count@=0\relax
      \def\@glo@thislettergrp{glssymbols}%
    \else
      \ifdefstring\@glo@sorttype{case}%
      {%
        \count@=#1\relax
      }%
      {%
      }%
    \edef\@glo@thislettergrp{\the\count@}%
    \fi
  }%
}
\newcommand{\@gls@noidx@do}[1]{%
  \global\letcs{\@gls@loclist}{glo\@glsdetoklabel{#1}@loclist}%
  \ifglshasparent{#1}%
  {%
    \gls@level=\csuse{glo\@glsdetoklabel{#1}@level}\relax
    \ifdefvoid{\@gls@loclist}
    {%
      \subglossentry{\gls@level}{#1}{}%
    }%
    {%
      \subglossentry{\gls@level}{#1}%
    }%
    \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
  }%
}

```

```

    }%
  }%
}%
{%
\letcs{\@gls@sort}{glo@glsdetoklabel{#1}@sort}%
\expandafter\glo@grabfirst\@gls@sort{}{}\@nil
\ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
{}%
{%
\ifdefempty{\@gls@currentlettergroup}{}%
{%
\global\let\@glo@thislettergrp\@glo@thislettergrp
\glsgroupskip
}%
\glsgroupheading{\@glo@thislettergrp}%
}%
\global\let\@gls@currentlettergroup\@glo@thislettergrp
\ifdefvoid{\@gls@loclist}
{%
\glossentry{#1}{}%
}%
{%
\glossentry{#1}%
{%
\glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
}%
}%
}%
}
\newcommand*{\glsnoidxloclist}[1]{%
\def\@gls@noidxloclist@sep{}%
\def\@gls@noidxloclist@prev{}%
\forlistloop{\glsnoidxloclisthandler}{#1}%
}
\newcommand*{\glsnoidxloclisthandler}[1]{%
\ifdefstring{\@gls@noidxloclist@prev}{#1}%
{%
}%
{%
\@gls@noidxloclist@sep
#1%
\def\@gls@noidxloclist@sep{\delimN}%
\def\@gls@noidxloclist@prev{#1}%
}%
}
\newcommand*{\glsnoidxdisplayloclisthandler}[1]{%
\ifdefstring{\@gls@noidxloclist@prev}{#1}%
{%
}%
{%
}

```

```

    \@gls@noidxloclist@sep
    \@gls@noidxloclist@prev
    \def\@gls@noidxloclist@prev{#1}%
  }%
}
\newcommand*\glsnoidxdisplayloc[4]{%
  \setentrycounter[#1]{#2}%
  \csuse{#3}{#4}%
}
\newcommand*\@gls@reference}[3]{%
  \glsdoifexistsorwarn{#2}%
  {%
    \ifcsundef{\@glsref@#1}{\csgdef{\@glsref@#1}{}}{%
      \ifinlistcs{#2}{\@glsref@#1}%
      {}%
      {\listcsgadd{\@glsref@#1}{#2}}%
      \ifcsundef{glo@\glsdetoklabel{#2}@loclist}%
      {\csgdef{glo@\glsdetoklabel{#2}@loclist}{}}%
      {}%
      \listcsgadd{glo@\glsdetoklabel{#2}@loclist}{#3}%
    }%
  }
}
\define@key{printgloss}{type}{\def\@glo@type{#1}}
\define@key{printgloss}{title}{%
  \def\glossarytitle{#1}%
  \let\gls@dotocitle\relax
}
\define@key{printgloss}{toctitle}{%
  \def\glossarytoctitle{#1}%
  \let\gls@dotocitle\relax
}
\define@key{printgloss}{style}{%
  \ifcsundef{\@glsstyle@#1}%
  {%
    \PackageError{glossaries}%
    {Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \def\@glossarystyle{\setglossentrycompatibility
      \csname @glsstyle@#1\endcsname}%
  }%
}
\define@choicekey{printgloss}{numberedsection}%
  [\@gls@numberedsection@val\@gls@numberedsection@nr]%
  {false,nolabel,autolabel,nameref}[nolabel]%
  {%
    \ifcase\@gls@numberedsection@nr\relax
      \renewcommand*\@@glossarysecstar{*}%
      \renewcommand*\@@glossaryseclabel{}%
    \or

```

```

\renewcommand*\@@glossarysecstar{}%
\renewcommand*\@@glossaryseclabel{}%
\or
\renewcommand*\@@glossarysecstar{}%
\renewcommand*\@@glossaryseclabel{\label{\glsautoprefix\@glo@type}}%
\or
\renewcommand*\@@glossarysecstar{*}%
\renewcommand*\@@glossaryseclabel{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}}%
\fi
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
\csuse{glsnogroupskip#1}}%
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
\csuse{glsnopostdot#1}}%
}
\newcommand*\GlsEntryCounterLabelPrefix{glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
\csuse{glsentrycounter#1}}%
\@gls@define@glossaryentrycounter
}
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
\csuse{glsesubentrycounter#1}}%
\@gls@define@glossarysubentrycounter
}
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
\ifglsnonumberlist
\def\glossaryentrynumbers##1{}%
\else
\def\glossaryentrynumbers##1{##1}%
\fi}
\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
\newcommand*\@glo@no@assign@sortkey[1]{%
\PackageError{glossaries}{‘sort’ key not permitted with
\string\printglossary}%
{The ‘sort’ key may only be used with \string\printnoidxglossary}%
}
\newcommand*\@glo@assign@sortkey[1]{%
\def\@glo@sorttype{#1}}%
}
\newcommand*\@glsnonextpages{%
\gdef\glossaryentrynumbers##1{%
\glsresetentrylist
}%
}
\newcommand*\@glsnextpages{%
\gdef\glossaryentrynumbers##1{%
##1\glsresetentrylist}}

```

```

\newcommand*\glsresetentrylist}{%
  \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
\newcommand*\glsnonextpages}{}
\newcommand*\glsnextpages}{}
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
\newcommand*\glsresetsubentrycounter}{%
  \ifglssubentrycounter
    \setcounter{glossarysubentry}{0}%
  \fi
}
\newcommand*\glsresetentrycounter}{%
  \ifglentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*\glsstepentry}[1]{%
  \ifglentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
\newcommand*\glsstepsubentry}[1]{%
  \ifglssubentrycounter
    \protected@edef\currentglssubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
  \fi
}
\newcommand*\glsrefentry}[1]{%
  \ifglentrycounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else
    \ifglssubentrycounter
      \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
      \gls{#1}%
    \fi
  \fi
}
\newcommand*\glsentrycounterlabel}{%
  \ifglentrycounter
    \theglossaryentry.\space
  \fi
}
\newcommand*\glssubentrycounterlabel}{%
  \ifglssubentrycounter
    \theglossarysubentry)\space
  \fi
}

```

```

\newcommand*{\glentryitem}[1]{%
  \ifglentrycounter
    \glstepentry{#1}\glentrycounterlabel
  \else
    \glresetsubentrycounter
  \fi
}
\newcommand*{\glssubentryitem}[1]{%
  \ifglssubentrycounter
    \glstepsubentry{#1}\glssubentrycounterlabel
  \fi
}
\ifcsundef{theglossary}%
{%
  \newenvironment{theglossary}{}{}%
}%
{%
  \@gls@warnontheglossdefined
  \renewenvironment{theglossary}{}{}%
}
\newcommand*{\glossaryheader}{}
\newcommand*{\glstarget}[2]{\@glstarget{\glo@linkprefix#1}{#2}}
\providecommand*{\compatibleglossentry}[2]{%
  \toks@{#2}%
  \protected@edef\@do@glossentry{\noexpand\glossaryentryfield{#1}%
    {\noexpand\glsnamefont
      {\expandafter\expandonce\csname glo@#1@name\endcsname}}%
    {\expandafter\expandonce\csname glo@#1@desc\endcsname}%
    {\expandafter\expandonce\csname glo@#1@symbol\endcsname}%
    {\the\toks@}}%
  }%
  \@do@glossentry
}
\newcommand*{\glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
    \expandafter\glsnamefont\expandafter{\glo@name}%
  }%
}
\newcommand*{\Glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsnamefont{\Glsentryname{#1}}%
  }%
}
\newcommand*{\glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glentrydesc{#1}%
  }%
}

```

```

    }%
  }
  \newcommand*\Glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \Glsentrydesc{#1}%
    }%
  }
  \newcommand*\glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \glsentrysymbol{#1}%
    }%
  }
  \newcommand*\Glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \Glsentrysymbol{#1}%
    }%
  }
  \providecommand*\compatiblesubglossentry}[3]{%
    \toks@{#3}%
    \protected@edef\@do@subglossentry{\noexpand\glossarysubentryfield{\number#1}%
    {#2}}%
    {\noexpand\glsnamefont
     {\expandafter\expandonce\csname glo@#2@name\endcsname}}%
    {\expandafter\expandonce\csname glo@#2@desc\endcsname}%
    {\expandafter\expandonce\csname glo@#2@symbol\endcsname}%
    {\the\toks@}%
  }%
  \@do@subglossentry
}
\newcommand*\setglossentrycompatibility{%
  \let\glossentry\compatibleglossentry
  \let\subglossentry\compatiblesubglossentry
}
\setglossentrycompatibility
\newcommand*\glossaryentryfield}[5]{%
  \GlossariesWarning
  {Deprecated use of \string\glossaryentryfield.^.^J
   I recommend you change to \string\glossentry.^.^J
   If you've just upgraded, try removing your gls auxiliary
   files^^J and recompile}%
  \noindent\textbf{\glstarget{#1}{#2}} #4 #3. #5\par}
\newcommand*\glossarysubentryfield}[6]{%
  \GlossariesWarning
  {Deprecated use of \string\glossarysubentryfield.^.^J
   I recommend you change to \string\subglossentry.^.^J
   If you've just upgraded, try removing your gls auxiliary
   files^^J and recompile}%

```



```

\glstarget{#2}{\strut}#4. #6\par}
\newcommand*{\glsgroupskip}{}
\newcommand*{\glsgroupheading}[1]{}
\newcommand*{\glsgetgrouptitle}[1]{%
  \@gls@getgrouptitle{#1}{\@gls@grptitle}%
  \@gls@grptitle
}
\newcommand*{\@gls@getgrouptitle}[2]{%
  \dtl@ifsingle{#1}%
  {%
    \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
  }%
  {%
    \ifboolexpr{test{\ifstrequal{#1}{glsymbols}}
      or test{\ifstrequal{#1}{glsnumbers}}}%
    {%
      \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
    }%
    {%
      \def#2{#1}%
    }%
  }%
}
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%
  \DTLifint{#1}%
  {\edef#2{\char#1\relax}}%
  {%
    \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
  }%
}
\newcommand*{\glsgetgrouplabel}[1]{%
\ifthenelse{\equal{#1}{\glsymbolsgroupname}}{glsymbols}{%
\ifthenelse{\equal{#1}{\glsnumbersgroupname}}{glsnumbers}{#1}}%
\newcommand*{\setentrycounter}[2][1]{%
  \def\@glo@counterprefix{#1}%
  \ifx\@glo@counterprefix\@empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1}%
  \fi
  \def\glsentrycounter{#2}%
}
\newcommand*{\setglossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \csname @glsstyle@#1\endcsname
  }%
}

```

```

\ifx\@glossary@default@style\relax
  \protected@edef\@glossary@default@style{#1}%
\fi
}
\newcommand*\@glossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \GlossariesWarning
    {Deprecated command \string\glossarystyle.^^J
    I recommend you switch to \string\setglossarystyle\space unless
    you want to maintain backward compatibility}%
    \setglossentrycompatibility
    \csname @glsstyle@#1\endcsname
    \ifcsdef{@glscompstyle@#1}%
    {\setglossentrycompatibility\cuse{@glscompstyle@#1}}%
    {}%
  }%
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\newcommand*\@newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
\newcommand*\@renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
\newcommand*\@glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\glsnumber#1{#1}%
}%
{%
  \def\glsnumber#1{\@glsnumber#1\nohyperpage{}}\@nil}
}

```

```

\def\@glshypernumber#1\nohyperpage#2#3\@nil{%
  \ifx\#1\%
  \else
    \@delimR#1\delimR\delimR\%
  \fi
  \ifx\#2\%
  \else
    #2%
  \fi
  \ifx\#3\%
  \else
    \@glshypernumber#3\@nil
  \fi
}
\def\@delimR#1\delimR #2\delimR #3\{%
\ifx\#2\%
  \@delimN{#1}%
\else
  \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
\fi}
\def\@delimN#1{\@delimN#1\delimN \delimN\}
\def\@delimN#1\delimN #2\delimN#3\{%
\ifx\#3\%
  \@gls@numberlink{#1}%
\else
  \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
\fi
}
\def\@gls@numberlink#1{%
\begingroup
\toks@={}%
\@gls@removespaces#1 \@nil
\endgroup}
\def\@gls@removespaces#1 #2\@nil{%
\toks@=\expandafter{\the\toks@#1}%
\ifx\#2\%
  \edef\@gls@x{\the\toks@}%
  \ifx\@gls@x\empty
  \else
    \hyperlink{\glsentrycounter\@glo@counterprefix\the\toks@}%
      {\the\toks@}%
  \fi
\else
  \@gls@ReturnAfterFi{%
    \@gls@removespaces#2\@nil
  }%
\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
\newcommand*\hyperrm}[1]{\textrm{\glshypernumber{#1}}}

```

```

\newcommand*\hypersf}[1]{\textsf{\glshypernumber{#1}}}
\newcommand*\hypertt}[1]{\texttt{\glshypernumber{#1}}}
\newcommand*\hyperbf}[1]{\textbf{\glshypernumber{#1}}}
\newcommand*\hypermd}[1]{\textmd{\glshypernumber{#1}}}
\newcommand*\hyperit}[1]{\textit{\glshypernumber{#1}}}
\newcommand*\hypersl}[1]{\textsl{\glshypernumber{#1}}}
\newcommand*\hyperup}[1]{\textup{\glshypernumber{#1}}}
\newcommand*\hypersc}[1]{\textsc{\glshypernumber{#1}}}
\newcommand*\hyperemph}[1]{\emph{\glshypernumber{#1}}}
\newcommand{\oldacronym}[4][\gls@label]{%
  \def\gls@label{#2}%
  \newacronym[#4]{#1}{#2}{#3}%
  \ifcsundef{xspace}%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}}%
    }%
  }%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
        \noexpand\gls{#1}\noexpand\xspace}%
      }%
    }%
  }
}
\newcommand{\newacronym}[4][[]]{
\newcommand*\acrpluralsuffix}{\glsacrpluralsuffix}
\newrobustcmd*\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
\newcommand*\glsshortkey}{short}
\newcommand*\glsshortpluralkey}{shortplural}
\newcommand*\glslongkey}{long}
\newcommand*\glslongpluralkey}{longplural}
\newrobustcmd*\acrfull}{\@gls@hyp@opt\ns@acrfull}
\newcommand*\ns@acrfull[2][[]]{%
  \new@ifnextchar[{\@acrfull{#1}{#2}}%
    {\@acrfull{#1}{#2}[]}%
}
\def\@acrfull#1#2[#3]{%
  \acrfullfmt{#1}{#2}{#3}%
}
\newcommand*\acrfullfmt}[3]{%
  \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newcommand{\acrlinkfullformat}[5]{%
  \acrfullformat{#1}{#3}{#4}{#5}{#2}{#3}{#4}[]}%
}
\newcommand{\acrfullformat}[2]{#1\glspace{#2}}
\newrobustcmd{\glspace}{\space}
\newrobustcmd*\Acrfull}{\@gls@hyp@opt\ns@Acrfull}
\newcommand*\ns@Acrfull[2][[]]{%

```

```

    \new@ifnextchar [{\@Acrfull{#1}{#2}}%
        {\@Acrfull{#1}{#2} []}%
}
\def\@Acrfull#1#2[#3]{%
    \Acrfullfmt{#1}{#2}{#3}%
}
\newcommand*\Acrfullfmt[3]{%
    \acrlinkfullformat{\@Acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newrobustcmd*\@ACRfull{\@gls@hyp@opt\ns@ACRfull}
\newcommand*\ns@ACRfull[2] []{%
    \new@ifnextchar [{\@ACRfull{#1}{#2}}%
        {\@ACRfull{#1}{#2} []}%
}
\def\@ACRfull#1#2[#3]{%
    \ACRfullfmt{#1}{#2}{#3}%
}
\newcommand*\ACRfullfmt[3]{%
    \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
\newrobustcmd*\@acrfullpl{\@gls@hyp@opt\ns@acrfullpl}
\newcommand*\ns@acrfullpl[2] []{%
    \new@ifnextchar [{\@acrfullpl{#1}{#2}}%
        {\@acrfullpl{#1}{#2} []}%
}
\def\@acrfullpl#1#2[#3]{%
    \acrfullplfmt{#1}{#2}{#3}%
}
\newcommand*\@acrfullplfmt[3]{%
    \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*\@Acrfullpl{\@gls@hyp@opt\ns@Acrfullpl}
\newcommand*\ns@Acrfullpl[2] []{%
    \new@ifnextchar [{\@Acrfullpl{#1}{#2}}%
        {\@Acrfullpl{#1}{#2} []}%
}
\def\@Acrfullpl#1#2[#3]{%
    \ACRfullplfmt{#1}{#2}{#3}%
}
}

```

```

\newcommand*\ACRfullplfmt}[3]{%
  \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand*\acrnameformat}[2]{\acronymfont{#1}}
\newtoks\glskeylisttok
\newtoks\glslabeltok
\newtoks\glsshorttok
\newtoks\glslongtok
\newcommand*\newacronymhook{}
\newcommand*\SetGenericNewAcronym{%
  \let\@Gls@entryname\@Gls@acrentryname
  \renewcommand{\newacronym}[4][]{%
    \ifdefempty{\@glsacronymlists}%
    {%
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
    }%
    {}%
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
  }%
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\expandonce{\acronymentry{##2}}},%
      sort={\acronymsort{\the\glsshorttok}{\the\glslongtok}},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      \GenericAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}%
\renewcommand*\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}%
}
\renewcommand*\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}%
}
\renewcommand*\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}%
}

```

```

\renewcommand*\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}%
\renewcommand*\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}%
\renewcommand*\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}%
\renewcommand*\glsentryfull}[1]{\genacrfullformat{##1}{}}%
\renewcommand*\Glsentryfull}[1]{\Genacrfullformat{##1}{}}%
\renewcommand*\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}%
\renewcommand*\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}%
}
\newcommand*\GenericAcronymFields{description={\the\glslongtok}}
\newcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
\newcommand*\acronymsort}[2]{##1}
\newcommand*\setacronymstyle}[1]{%
  \ifcsundef{@glsacr@dispstyle@#1}
  {%
    \PackageError{glossaries}{Undefined acronym style ‘#1’}{%
  }%
  {%
    \ifdefempty{\@glsacronymlists}%
    {%
      \DeclareAcronymList{acronymtype}%
    }%
    {}%
    \SetGenericNewAcronym
    \GlsUseAcrStyleDefs{##1}%
    \@for\@gls@type:=\@glsacronymlists\do{%
      \defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{##1}}%
    }%
  }%
}
\newcommand*\newacronymstyle}[3]{%
  \ifcsdef{@glsacr@dispstyle@#1}%
  {%
    \PackageError{glossaries}{Acronym style ‘#1’ already exists}{%
  }%
  {%
    \csdef{@glsacr@dispstyle@#1}{##2}%
    \csdef{@glsacr@styledefs@#1}{##3}%
  }%
}
\newcommand*\renewacronymstyle}[3]{%
  \ifcsdef{@glsacr@dispstyle@#1}%
  {%
    \csdef{@glsacr@dispstyle@#1}{##2}%
    \csdef{@glsacr@styledefs@#1}{##3}%
  }%
  {%

```

```

    \PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{%
  }%
}
\newcommand*\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
{%
  \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
  \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
  \renewcommand*\genacrfullformat}[2]{%
    \glsentrylong{##1}##2\space
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*\Genacrfullformat}[2]{%
    \Glsentrylong{##1}##2\space
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*\genplacrfullformat}[2]{%
    \glsentrylongpl{##1}##2\space
    (\protect\firstacronymfont{\glsentryshortpl{##1}})%
  }%
  \renewcommand*\Genplacrfullformat}[2]{%
    \Glsentrylongpl{##1}##2\space
    (\protect\firstacronymfont{\glsentryshortpl{##1}})%
  }%
  \renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
  \renewcommand*\acronymsort}[2]{##1}%
  \renewcommand*\acronymfont}[1]{##1}%
  \renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
  \renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
  \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
  \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
  \renewcommand*\genacrfullformat}[2]{%
    \glsentrylong{##1}##2\glsacsacspace{##1}%
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*\Genacrfullformat}[2]{%
    \Glsentrylong{##1}##2\glsacsacspace{##1}%
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*\genplacrfullformat}[2]{%
    \glsentrylongpl{##1}##2\glsacsacspace{##1}%
    (\protect\firstacronymfont{\glsentryshortpl{##1}})%
  }%

```



```

}%
\renewcommand*\Genplacrfullformat}[2]{%
  \Glsentrylongpl{##1}##2\glsacspace{##1}%
  (\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newcommand*\glsacspace}[1]{%
  \settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{##1}})}%
  \ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{
  \ifglsashaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshort{##1}}##2\space
  (\glsentrylong{##1})%
}%
\renewcommand*\Genacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshort{##1}}##2\space
  (\glsentrylong{##1})%
}%
\renewcommand*\genplacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
  (\glsentrylongpl{##1})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
  (\glsentrylongpl{##1})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*\acronymfont}[1]{\textsc{##1}}%

```

```

    \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
  }
  \newacronymstyle{long-sm-short}%
  {%
    \GlsUseAcrEntryDispStyle{long-short}%
  }%
  {%
    \GlsUseAcrStyleDefs{long-short}%
    \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
    \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
  }
  \newacronymstyle{sc-short-long}%
  {%
    \GlsUseAcrEntryDispStyle{short-long}%
  }%
  {%
    \GlsUseAcrStyleDefs{short-long}%
    \renewcommand{\acronymfont}[1]{\textsc{##1}}%
    \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
  }
  \newacronymstyle{sm-short-long}%
  {%
    \GlsUseAcrEntryDispStyle{short-long}%
  }%
  {%
    \GlsUseAcrStyleDefs{short-long}%
    \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
    \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
  }
  \newacronymstyle{long-short-desc}%
  {%
    \GlsUseAcrEntryDispStyle{long-short}%
  }%
  {%
    \GlsUseAcrStyleDefs{long-short}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymsort}[2]{##2}%
    \renewcommand*{\acronymentry}[1]{%
      \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
  }
  \newacronymstyle{long-sp-short-desc}%
  {%
    \GlsUseAcrEntryDispStyle{long-sp-short}%
  }%
  {%
    \GlsUseAcrStyleDefs{long-sp-short}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymsort}[2]{##2}%
    \renewcommand*{\acronymentry}[1]{%
      \glsentrylong{##1}\glsacspace{##1}(\acronymfont{\glsentryshort{##1}})}%
  }

```

```

}
\newacronymstyle{long-sc-short-desc}%
{
  \GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{
  \GlsUseAcrStyleDefs{long-sc-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sm-short-desc}%
{
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{short-long-desc}%
{
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{sc-short-long-desc}%
{
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{sm-short-long-desc}%
{
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{

```

```

\GlsUseAcrStyleDefs{sm-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \Glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \mfirstucMakeUppercase
            {\glsentrylongpl{\glslabel}\glsinsert}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \Glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \mfirstucMakeUppercase
          {\glsentrylong{\glslabel}\glsinsert}%
      }%
    }%
  }%
  }%
  {%
    \glsgenentryfmt
  }%
  {\glscustomtext\glsinsert}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\acrfullfmt}[3]{%
    \glslink[##1]{##2}{\glsentrylong{##2}##3\space

```

```

        (\acronymfont{\glsentryshort{##2}})}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Glsentrylong{##2}##3\space
    (\acronymfont{\glsentryshort{##2}})}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\glsentrylong{##2}##3\space
      (\acronymfont{\glsentryshort{##2}})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\glsentrylongpl{##2}##3\space
    (\acronymfont{\glsentryshortpl{##2}})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\glsentrylongpl{##2}##3\space
      (\acronymfont{\glsentryshortpl{##2}})}}}%
\renewcommand*{\glsentryfull}[1]{%
  \glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
  \glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
  \GlsUseAcrEntryDispStyle{dua}%
}%
{%
  \GlsUseAcrStyleDefs{dua}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentrylong{##1}}}%
  \renewcommand*{\acronymsort}[2]{##2}%
}%
\newacronymstyle{footnote}%
{%
  \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%

```

```

\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\glsHYperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshort{##1}}##2%
  \protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
  \firstacronymfont{\Glsentryshort{##1}}##2%
  \protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
  \mfirstucMakeUppercase{\acronymfont{\glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
  \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}}}%
\renewcommand*{\glsentryfull}[1]{%
  \acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\glsentryfullpl}[1]{%
  \acronymfont{\glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%

```

```

}
\newacronymstyle{footnote-sc}%
{
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}%
\newacronymstyle{footnote-sm}%
{
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}%
\newacronymstyle{footnote-desc}%
{
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{footnote-sc-desc}%
{
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{footnote-sm-desc}%
{
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}%
{
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*{\GenericAcronymFields}{}%
}

```

```

\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%
  \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newcommand*\DefineAcronymSynonyms){%
  \let\acs\acrshort
  \let\Acs\Acrshort
  \let\acsp\acrshortpl
  \let\Acsp\Acrshortpl
  \let\acl\acrlong
  \let\aclp\acrlongpl
  \let\Acl\Acrlong
  \let\Aclp\Acrlongpl
  \let\acf\acrfull
  \let\acfp\acrfullpl
  \let\Acf\Acrfull
  \let\Acfp\Acrfullpl
  \let\ac\gls
  \let\Ac\Gls
  \let\acp\glspl
  \let\Acp\Glspl
}
\ifglsacrshortcuts
  \DefineAcronymSynonyms
\fi
\newcommand*\SetDefaultAcronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*\DefaultNewAcronymDef){%
  \edef\do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      sort={\the\glsshorttok},%
      text={\the\glsshorttok},%
      first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
      plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      firstplural={\acrfullformat{\noexpand\expandonce\noexpand\@glo@longpl}%
        {\noexpand\expandonce\noexpand\@glo@shortpl}},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      \the\glskeylisttok
    }%
  }%
\let\@org@gls@assign@firstpl\gls@assign@firstpl

```



```

\let\@org@gl@assign@plural\gl@assign@plural
\let\@org@gl@assign@descplural\gl@assign@descplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@gl@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\@org@gl@assign@firstpl
\let\gl@assign@plural\@org@gl@assign@plural
\let\gl@assign@symbolplural\@org@gl@assign@symbolplural
\@gl@noexpand@assign@default@plurals
}
\newcommand*{\@gl@noexpand@assign@default@plurals}{%
  \expandafter\ifx\csname glo@the\gl@labeltok @plural\endcsname\@gl@@default@acr@plural
    \csxdef{glo@the\gl@labeltok @plural}{\expandonce\@glo@shortpl}%
  \fi
  \expandafter\ifx\csname glo@the\gl@labeltok @firstpl\endcsname\@gl@@default@acr@firstplural
    \csxdef{glo@the\gl@labeltok @firstpl}{%
      \expandonce\@glo@longpl\glsspace(\expandonce\@glo@shortpl)}%
    \fi
  \expandafter\ifx\csname glo@the\gl@labeltok @descplural\endcsname\@gl@@default@acr@descplural
    \csxdef{glo@the\gl@labeltok @descplural}{\expandonce\@glo@longpl}%
    \fi
}
\newcommand*{\@gl@@default@acr@plural}{\expandonce\@glo@shortpl}
\newcommand*{\@gl@@default@acr@firstplural}{%
  \expandonce\@glo@longpl\glsspace(\expandonce\@glo@shortpl)%
}
\newcommand*{\@gl@@default@acr@descplural}{\expandonce\@glo@longpl}
\newcommand*{\SetDefaultAcronymStyle}{%
  \@for\@gl@type:=\@gl@sacronymlists\do{%
    \SetDefaultAcronymDisplayStyle{\@gl@type}%
  }%
}
\renewcommand{\newacronym}[4][ ]{%
  \ifx\@gl@sacronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetDefaultAcronymDisplayStyle{\@glo@type}%
  \fi
  \gl@keylisttok{##1}%
  \gl@labeltok{##2}%
  \gl@shorttok{##3}%
  \gl@longtok{##4}%
  \newacronymhook
}

```

```

\DefaultNewAcronymDef
}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
\newcommand*{\acrlinkfootnote}[3]{%
\footnote{\glslink[#1]{#2}{#3}}%
}
\newcommand*{\acrnolinkfootnote}[3]{%
\footnote{#3}%
}
\newcommand*{\SetDescriptionFootnoteAcronymDisplayStyle}[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\ifglsused{\glslabel}%
{%
\acronymfont{\glsgenentryfmt}%
}%
{%
\firstacronymfont{\glsgenentryfmt}%
\ifgls hassymbol{\glslabel}%
{%
\expandafter\protect\expandafter\acrfootnote\expandafter
{\@gls@link@opts}{\@gls@link@label}%
{%
\glsifplural
{\glsentrysymbolplural{\glslabel}}%
{\glsentrysymbol{\glslabel}}%
}%
}%
}%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*{\DescriptionFootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
first={\the\glsshorttok},%
firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
}
}

```

```

        longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
        symbol={\the\glslongtok},%
        symbolplural={\noexpand\expandonce\noexpand\@glo@longpl},%
        \the\glskeylisttok
    }%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
    \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
    \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
    \@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionFootnoteAcronymStyle}{%
    \ifglsacrsmallcaps
        \@gls@deprecated@acr@warning{footnote,smallcaps,description}%
        {footnote-sc-desc}{short-sc-footnote-desc}%
    \else
        \ifglsacrsmaller
            \@gls@deprecated@acr@warning{footnote,smaller,description}%
            {footnote-sm-desc}{short-sm-footnote-desc}%
        \else
            \@gls@deprecated@acr@warning{footnote,description}%
            {footnote-desc}{short-footnote-desc}%
        \fi
    \fi
\renewcommand{\newacronym}[4][]{%
    \ifx\@glsacronymlists\@empty
        \def\@glo@type{\acronymtype}%
        \setkeys{glossentry}{##1}%
        \DeclareAcronymList{\@glo@type}%
        \SetDescriptionFootnoteAcronymDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \DescriptionFootnoteNewAcronymDef
}%

```

```

\@for\@gls@type:=\@glsacronymlists\do{%
  \SetDescriptionFootnoteAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
  \renewcommand*\acronymfont[1]{\textsc{##1}}%
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
\else
  \ifglsacrsmaller
    \renewcommand*\acronymfont[1]{\textsmaller{##1}}%
  \fi
\fi
\ifglsacrdua
  \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
can’t both be set}{}%
\fi
}%
\newcommand*\SetDescriptionDUAAcronymDisplayStyle[1]{%
  \defglsentryfmt[##1]{\glsgenentryfmt}%
}
\newcommand*\DescriptionDUANewAcronymDef{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glslongtok},%
      sort={\the\glslongtok},%
      text={\the\glslongtok},%
      first={\the\glslongtok},%
      plural={\noexpand\expandonce\noexpand\@glo@longpl},%
      firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      symbol={\the\glsshorttok},%
      symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      \the\glskeylisttok
    }%
  }%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}

```

```

}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*\SetDescriptionDUAAcronymStyle{%
\ifglsacrsmallcaps
\PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
can’t both be set}{}%
\else
\ifglsacrsmaller
\PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
can’t both be set}{}%
\fi
\fi
\@gls@deprecated@acr@warning{dua,description}{dua-desc}{long-noshort-desc}%
\renewcommand{\newacronym}[4][]{%
\ifx\@glsacronymlists\@empty
\def\@glo@type{\acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@glo@type}%
\SetDescriptionDUAAcronymDisplayStyle{\@glo@type}%
\fi
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\DescriptionDUANewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetDescriptionDUAAcronymDisplayStyle{\@gls@type}%
}%
}%
\newcommand*\SetDescriptionAcronymDisplayStyle[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\ifglsused{\glslabel}%
{%
\let\gls@org@insert\glsinsert
\let\glsinsert\@empty
\acronymfont{\glsgenentryfmt}\gls@org@insert
}%
}%
\glsgenentryfmt
\ifglshassymbol{\glslabel}%
{%
\glsifplural

```

```

        {%
        \def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%
        }%
        {%
        \def\@glo@symbol{\glsentrysymbol{\glslabel}}%
        }%
        \space(\protect\firstacronymfont
        {\gls caps case
        {\@glo@symbol}
        {\@glo@symbol}
        {\mfirstucMakeUppercase{\@glo@symbol}}})%
        }%
        {}%
    }%
    }%
    {\gls custom text\gls insert}%
}
}
\newcommand*{\DescriptionNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand
\acrnameformat{\the\gls shorttok}{\the\gls longtok}},%
sort={\the\gls shorttok},%
first={\the\gls longtok},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
text={\the\gls shorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\gls shorttok},%
shortplural={\the\gls shorttok\noexpand\acrpluralsuffix},%
long={\the\gls longtok},%
longplural={\the\gls longtok\noexpand\acrpluralsuffix},%
symbol={\noexpand\@glo@text},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\gls keylisttok}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
\@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
\@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
\@gls@expand@field{##1}{symbolplural}{##2}%
}%
}

```

```

\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionAcronymStyle}{%
\ifglsacrsmallcaps
\@gls@deprecated@acr@warning{smallcaps,description}%
{long-sc-short-desc}{long-short-sc-desc}%
\else
\ifglsacrsmaller
\@gls@deprecated@acr@warning{smaller,description}%
{long-sm-short-desc}{long-short-sm-desc}%
\else
\@gls@deprecated@acr@warning{description}%
{long-short-desc}{long-short-desc}%
\fi
\fi
\renewcommand{\newacronym}[4][ ]{%
\ifx\@glsacronymlists\@empty
\def\@glo@type{\acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@glo@type}%
\SetDescriptionAcronymDisplayStyle{\@glo@type}%
\fi
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\DescriptionNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetDescriptionAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
\renewcommand{\acronymfont}[1]{\textsc{##1}}
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
\else
\ifglsacrsmaller
\renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
\fi
\fi
}%
\newcommand*{\SetFootnoteAcronymDisplayStyle}[1]{%
\defglsentryfmt[##1]{%
\ifdefempty\glscustomtext
{%
\let\gls@org@insert\glsinsert
\let\glsinsert\@empty

```

```

\ifglsused{\glslabel}%
{%
  \acronymfont{\glsgenentryfmt}\gls@org@insert
}%
{%
  \firstacronymfont{\glsgenentryfmt}\gls@org@insert
  \ifglsashaslong{\glslabel}%
  {%
    \expandafter\protect\expandafter\acrfootnote\expandafter
    {\@gls@link@opts}{\@gls@link@label}%
    {%
      \glsifplural
        {\glsentrylongpl{\glslabel}}%
        {\glsentrylong{\glslabel}}%
    }%
  }%
  }%
  {}%
}%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*{\FootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\glsshorttok}},%
    sort={\the\glsshorttok},%
    text={\the\glsshorttok},%
    plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
    first={\the\glsshorttok},%
    firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
    short={\the\glsshorttok},%
    shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
    long={\the\gslongtok},%
    longplural={\the\gslongtok\noexpand\acrpluralsuffix},%
    description={\the\gslongtok},%
    descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
    \the\glskeylisttok
  }%
}%
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}

```



```

}%
\def\gls@assign@descplural##1##2{%
  \@gls@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@descplural\@org@gls@assign@descplural
}
\newcommand*\SetFootnoteAcronymStyle{%
  \ifglsacrsmallcaps
    \@gls@deprecated@acr@warning{footnote,smallcaps}{footnote-sc}{short-sc-footnote}%
  \else
    \ifglsacrsmaller
      \@gls@deprecated@acr@warning{footnote,smaller}{footnote-sm}{short-sm-footnote}%
    \else
      \@gls@deprecated@acr@warning{footnote}{footnote}{short-footnote}%
    \fi
  \fi
\renewcommand{\newacronym}[4] []{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetFootnoteAcronymDisplayStyle{\@glo@type}%
  \fi
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \glslongtok{##4}%
  \newacronymhook
  \FootnoteNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
  \SetFootnoteAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
  \renewcommand*\acronymfont[1]{\textsc{##1}}%
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
\else
  \ifglsacrsmaller
    \renewcommand*\acronymfont[1]{\textsmaller{##1}}%
  \fi
\fi
\ifglsacrdua
  \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
  can’t both be set}{}%
\fi
}%
\DeclareRobustCommand*\glsdoparenifnotempty}[2]{%

```

```

\protected@edef\gls@tmp{#1}%
\ifdefempty\gls@tmp
{}%
{%
  \ifx\gls@tmp\@gls@default@value
  \else
  \space (#2{#1})%
  \fi
}%
}
\newcommand*{\SetSmallAcronymDisplayStyle}[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\gls@customtext
{%
  \let\gls@org@insert\gls@insert
  \let\gls@insert\@empty
  \ifglsused{\gls@label}%
  {%
    \acronymfont{\gls@genentryfmt}\gls@org@insert
  }%
  {%
    \gls@genentryfmt
    \ifgls@shassymbol{\gls@label}%
    {%
      \gls@ifplural
      {%
        \def\@glo@symbol{\gls@entrysymbolplural{\gls@label}}%
      }%
      {%
        \def\@glo@symbol{\gls@entrysymbol{\gls@label}}%
      }%
      \space
      (\gls@scapscase
      {\firstacronymfont{\@glo@symbol}}%
      {\firstacronymfont{\@glo@symbol}}%
      {\firstacronymfont{\mfirstucMakeUppercase{\@glo@symbol}}})%
    }%
    {}%
  }%
  }%
  {\gls@customtext\gls@insert}%
}%
}
\newcommand*{\SmallNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\gls@labeltok}%
{%
  type=\acronymtype,%
  name={\noexpand\acronymfont{\the\gls@shorttok}},%
  sort={\the\gls@shorttok},%

```

```

text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
first={\the\glslongtok},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\noexpand\@glo@first},%
descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
symbol={\the\glsshorttok},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@gls@expand@field{##1}{descplural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@descplural\@org@gls@assign@descplural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetSmallAcronymStyle}{%
  \ifglsacrsmallcaps
    \gls@deprecated@acr@warning{smallcaps}{long-sc-short}{long-short-sc}%
  \else
    \gls@deprecated@acr@warning{smaller}{long-sm-short}{long-short-sm}%
  \fi
\renewcommand{\newacronym}[4][ ]{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetSmallAcronymDisplayStyle{\@glo@type}%
  \fi
}

```

```

\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\SmallNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
  \SetSmallAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
  \renewcommand*\acronymfont}[1]{\textsc{##1}}
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
\else
  \renewcommand*\acronymfont}[1]{\textsmaller{##1}}
\fi
\ifglsacrdua
  \ifglsacrsmallcaps
    \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
      can’t both be set}{}%
  \else
    \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
      can’t both be set}{}%
  \fi
\fi
}%
\newcommand*\SetDUADisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*\DUANewAcronymDef{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      text={\the\glslongtok},%
      first={\the\glslongtok},%
      plural={\noexpand\expandonce\noexpand\@glo@longpl},%
      firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      symbol={\the\glsshorttok},%
      symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      \the\glskeylisttok
    }%
  }%
}%

```

```

\let\@org@gl@assign@firstpl\gl@assign@firstpl
\let\@org@gl@assign@plural\gl@assign@plural
\let\@org@gl@assign@symbolplural\gl@assign@symbolplural
\let\@org@gl@assign@descplural\gl@assign@descplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@symbolplural##1##2{%
  \@gl@expand@field{##1}{symbolplural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@gl@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\@org@gl@assign@firstpl
\let\gl@assign@plural\@org@gl@assign@plural
\let\gl@assign@symbolplural\@org@gl@assign@symbolplural
\let\gl@assign@descplural\@org@gl@assign@descplural
}
\newcommand*\@SetDUASStyle{%
  \@gl@deprecated@acr@warning{dua}{dua}{long-noshort}%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@gl@acronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetDUADisplayStyle{\@glo@type}%
    \fi
    \gl@keylisttok{##1}%
    \gl@labeltok{##2}%
    \gl@shorttok{##3}%
    \gl@longtok{##4}%
    \newacronymhook
    \DUANewAcronymDef
  }%
  \@for\@gl@type:=\@gl@acronymlists\do{%
    \SetDUADisplayStyle{\@gl@type}%
  }%
}
\newcommand*\@gl@deprecated@acr@warning}[3]{%
  \GlossariesWarning{Acronym style option #1 deprecated.
  Use \string\setacronymstyle{#2} with just glossaries.sty or
  use \string\setabbreviationstyle[acronym]{#3} with
  glossaries-extra.sty}%
}
\newcommand*\@SetAcronymStyle{%
  \SetDefaultAcronymStyle

```

```

\ifglsacrdescription
  \ifglsacrfootnote
    \SetDescriptionFootnoteAcronymStyle
  \else
    \ifglsacrdua
      \SetDescriptionDUAAcronymStyle
    \else
      \SetDescriptionAcronymStyle
    \fi
  \fi
\else
  \ifglsacrfootnote
    \SetFootnoteAcronymStyle
  \else
    \ifthenelse{\boolean{glsacrsmallcaps}\OR
      \boolean{glsacrsmaller}}{%
      \SetSmallAcronymStyle
    }%
    \ifglsacrdua
      \SetDUASyle
    \fi
  }%
\fi
\fi
}
\SetAcronymStyle
\newcommand*{\SetCustomDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*{\CustomAcronymFields}{%
  name={\the\glsshorttok},%
  description={\the\glslongtok},%
  first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
  firstplural={\acrfullformat
    {\noexpand\glsentrylongpl{\the\glslabeltok}}%
    {\noexpand\glsentryshortpl{\the\glslabeltok}}},%
  text={\the\glsshorttok},%
  plural={\the\glsshorttok\noexpand\acrpluralsuffix}%
}
\newcommand*{\CustomNewAcronymDef}{%
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      short={\the\glsshorttok},%
      shortplural={\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},%
        \CustomAcronymFields,%
        \the\glskeylisttok
    }%
}%
\do@newglossaryentry
}
\newcommand*{\SetCustomStyle}{%
\renewcommand{\newacronym}[4][{}]{%
\ifx\@glsacronymlists\@empty
\def\@gls@type{\acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@gls@type}%
\SetCustomDisplayStyle{\@gls@type}%
\fi
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\CustomNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetCustomDisplayStyle{\@gls@type}%
}%
}
\RequirePackage{glossary-hypernav}[=v4.49]
\@gls@loadlist
\@gls@loadlong
\@gls@loadsuper
\@gls@loadtree
\ifx\@glossary@default@style\relax
\else
\setglossarystyle{\@glossary@default@style}
\fi
\newcommand*{\showgloparent}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
}
\newcommand*{\showglolevel}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
}
\newcommand*{\showglotext}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
}
\newcommand*{\showgloplural}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@plural\endcsname
}
}

```

```

\newcommand*\showglofirst}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@first\endcsname
}
\newcommand*\showglofirstpl}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@firstpl\endcsname
}
\newcommand*\showgloftype}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@ftype\endcsname
}
\newcommand*\showglocounter}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@counter\endcsname
}
\newcommand*\showglouser}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@user\endcsname
}
\newcommand*\showglouserii}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@userii\endcsname
}
\newcommand*\showglouseriii}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@useriii\endcsname
}
\newcommand*\showglouseriv}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@useriv\endcsname
}
\newcommand*\showglouserv}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@userv\endcsname
}
\newcommand*\showglouservi}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@uservi\endcsname
}
\newcommand*\showglongame}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@name\endcsname
}
\newcommand*\showglodesc}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@desc\endcsname
}
\newcommand*\showglodescplural}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@descplural\endcsname
}
\newcommand*\showglosort}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@sort\endcsname
}
\newcommand*\showglosymbol}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@symbol\endcsname
}
\newcommand*\showglosymbolplural}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@symbolplural\endcsname
}
\newcommand*\showgloshort}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@short\endcsname
}

```



```

}
\newcommand*\showglolong}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname
}
\newcommand*\showgloindex}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname
}
\newcommand*\showgloflag}[1]{%
  \expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname
}
\newcommand*\showglooclist}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@oclist\endcsname
}
\newcommand*\showglofield}[2]{%
  \csshow{glo@\glsdetoklabel{#1}@#2}%
}
\newcommand*\showacronymlists}{%
  \show\@glsacronymlists
}
\newcommand*\showglossaries}{%
  \show\@glo@types
}
\newcommand*\showglossaryin}[1]{%
  \expandafter\show\csname @glo@type@#1@in\endcsname
}
\newcommand*\showglossaryout}[1]{%
  \expandafter\show\csname @glo@type@#1@out\endcsname
}
\newcommand*\showglossarytitle}[1]{%
  \expandafter\show\csname @glo@type@#1@title\endcsname
}
\newcommand*\showglossarycounter}[1]{%
  \expandafter\show\csname @glo@type@#1@counter\endcsname
}
\newcommand*\showglossaryentries}[1]{%
  \expandafter\show\csname glist@#1\endcsname
}
\csname ifglscompatible-2.07\endcsname
  \RequirePackage{glossaries-compatible-207}[v4.49]
\fi

```

8.19 Rollback v4.49 (glossaries-accsupp-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-accsupp}[2021/11/01 v4.49 (NLCT)
  Experimental glossaries accessibility]
\DeclareOption*\PassOptionsToPackage{\CurrentOption}{glossaries}}
\ProcessOptions
\@ifpackageloaded{glossaries-extra}

```

```

{%
\ifx\@glsxtr@doaccsupp\empty
\GlossariesWarning{The 'glossaries-accsupp'
package has been loaded\MessageBreak
after the 'glossaries-extra' package. This\MessageBreak
can cause a failure to integrate both packages. \MessageBreak
Either use the 'accsupp' option when you load\MessageBreak
'glossaries-extra' or load 'glossaries-accsupp'\MessageBreak
before loading 'glossaries-extra'}%
\fi
}
{}
\def\compatibleglossentry#1#2{%
\toks@{#2}%
\protected@edef\@do@glossentry{%
\noexpand\accsuppglossaryentryfield{#1}%
{ \noexpand\glsnamefont
{\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@name\endcsname}}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@desc\endcsname}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@symbol\endcsname}%
{\the\toks@}%
}%
\@do@glossentry
}
\def\compatiblesubglossentry#1#2#3{%
\toks@{#3}%
\protected@edef\@do@subglossentry{%
\noexpand\accsuppglossarysubentryfield{\number#1}%
{#2}%
{ \noexpand\glsnamefont
{\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@name\endcsname}}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@desc\endcsname}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@symbol\endcsname}%
{\the\toks@}%
}%
\@do@subglossentry
}
\RequirePackage{glossaries}[v4.49]
\providecommand{\gls@accsupp@engine}{accsupp}
\providecommand{\gls@accessibility}[4]{#4}
\ifdefstring\gls@accsupp@engine{accsupp}
{
\RequirePackage{accsupp}
\renewcommand{\gls@accessibility}[4]{%
\BeginAccSupp{#1,#2={#3}}#4\EndAccSupp{}}
}
}
{}
\newcommand{\glsaccessibility}[4] [] {%
\@glsshowaccsupp{#1}{#2}{#3}%

```

```

\gls@accessibility{#1}{#2}{#3}{#4}%
}
\define@key{glossentry}{access}{%
  \def\@glo@access{#1}%
}
\define@key{glossentry}{textaccess}{%
  \def\@glo@textaccess{#1}%
}
\define@key{glossentry}{firstaccess}{%
  \def\@glo@firstaccess{#1}%
}
\define@key{glossentry}{pluralaccess}{%
  \def\@glo@pluralaccess{#1}%
}
\define@key{glossentry}{firstpluralaccess}{%
  \def\@glo@firstpluralaccess{#1}%
}
\define@key{glossentry}{symbolaccess}{%
  \def\@glo@symbolaccess{#1}%
}
\define@key{glossentry}{symbolpluralaccess}{%
  \def\@glo@symbolpluralaccess{#1}%
}
\define@key{glossentry}{descriptionaccess}{%
  \def\@glo@descaccess{#1}%
}
\define@key{glossentry}{descriptionpluralaccess}{%
  \def\@glo@descpluralaccess{#1}%
}
\define@key{glossentry}{shortaccess}{%
  \def\@glo@shortaccess{#1}%
}
\define@key{glossentry}{shortpluralaccess}{%
  \def\@glo@shortpluralaccess{#1}%
}
\define@key{glossentry}{longaccess}{%
  \def\@glo@longaccess{#1}%
}
\define@key{glossentry}{longpluralaccess}{%
  \def\@glo@longpluralaccess{#1}%
}
\define@key{glossentry}{user1access}{%
  \def\@glo@useriaccess{#1}%
}
\define@key{glossentry}{user2access}{%
  \def\@glo@useriiaccess{#1}%
}
\define@key{glossentry}{user3access}{%
  \def\@glo@useriiiaccess{#1}%
}
}

```

```

\define@key{glossentry}{user4access}{%
  \def\@glo@userivaccess{#1}%
}
\define@key{glossentry}{user5access}{%
  \def\@glo@uservaccess{#1}%
}
\define@key{glossentry}{user6access}{%
  \def\@glo@userviaccess{#1}%
}
\appto\@gls@keymap{,%
  {access}{access},%
  {textaccess}{textaccess},%
  {firstaccess}{firstaccess},%
  {pluralaccess}{pluralaccess},%
  {firstpluralaccess}{firstpluralaccess},%
  {symbolaccess}{symbolaccess},%
  {symbolpluralaccess}{symbolpluralaccess},%
  {descaccess}{descaccess},%
  {descpluralaccess}{descpluralaccess},%
  {shortaccess}{shortaccess},%
  {shortpluralaccess}{shortpluralaccess},%
  {longaccess}{longaccess},%
  {longpluralaccess}{longpluralaccess},%
  {user1access}{useriaccess},%
  {user2access}{useriiaccess},%
  {user3access}{useriiiaccess},%
  {user4access}{userivaccess},%
  {user5access}{uservaccess},%
  {user6access}{userviaccess}%
}
\def\@gls@noaccess{\relax}
\let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
\renewcommand*\@newglossaryentryprehook{%
  \@gls@oldnewglossaryentryprehook
  \def\@glo@access{\relax}%
  \def\@glo@textaccess{\@glo@access}%
  \def\@glo@firstaccess{\@glo@access}%
  \def\@glo@pluralaccess{\@glo@textaccess}%
  \def\@glo@firstpluralaccess{\@glo@pluralaccess}%
  \def\@glo@symbolaccess{\relax}%
  \def\@glo@symbolpluralaccess{\@glo@symbolaccess}%
  \def\@glo@descaccess{\relax}%
  \def\@glo@descpluralaccess{\@glo@descaccess}%
  \def\@glo@shortaccess{\relax}%
  \def\@glo@shortpluralaccess{\@glo@shortaccess}%
  \def\@glo@longaccess{\relax}%
  \def\@glo@longpluralaccess{\@glo@longaccess}%
  \def\@glo@useriaccess{\relax}%
  \def\@glo@useriiaccess{\relax}%
  \def\@glo@useriiiaccess{\relax}%
}

```

```

\def@glo@userivaccess{\relax}%
\def@glo@uservaccess{\relax}%
\def@glo@userviaccess{\relax}%
}
\let@gls@oldnewglossaryentryposthook@newglossaryentryposthook
\renewcommand*{@newglossaryentryposthook}{%
  @gls@oldnewglossaryentryposthook
  \expandafter
  \protected@xdef\csname glo@\@glo@label @access\endcsname{%
    \glo@access}%
  \expandafter
  \protected@xdef\csname glo@\@glo@label @textaccess\endcsname{%
    \glo@textaccess}%
  \expandafter
  \protected@xdef\csname glo@\@glo@label @firstaccess\endcsname{%
    \glo@firstaccess}%
  \expandafter
  \protected@xdef\csname glo@\@glo@label @pluralaccess\endcsname{%
    \glo@pluralaccess}%
  \expandafter
  \protected@xdef\csname glo@\@glo@label @firstpluralaccess\endcsname{%
    \glo@firstpluralaccess}%
  \expandafter
  \protected@xdef\csname glo@\@glo@label @symbolaccess\endcsname{%
    \glo@symbolaccess}%
  \expandafter
  \protected@xdef\csname glo@\@glo@label @symbolpluralaccess\endcsname{%
    \glo@symbolpluralaccess}%
  \expandafter
  \protected@xdef\csname glo@\@glo@label @descaccess\endcsname{%
    \glo@descaccess}%
  \expandafter
  \protected@xdef\csname glo@\@glo@label @descpluralaccess\endcsname{%
    \glo@descpluralaccess}%
  \expandafter
  \protected@xdef\csname glo@\@glo@label @shortaccess\endcsname{%
    \glo@shortaccess}%
  \expandafter
  \protected@xdef\csname glo@\@glo@label @shortpluralaccess\endcsname{%
    \glo@shortpluralaccess}%
  \expandafter
  \protected@xdef\csname glo@\@glo@label @longaccess\endcsname{%
    \glo@longaccess}%
  \expandafter
  \protected@xdef\csname glo@\@glo@label @longpluralaccess\endcsname{%
    \glo@longpluralaccess}%
  \expandafter
  \protected@xdef\csname glo@\@glo@label @useriaccess\endcsname{%
    \glo@useriaccess}%
  \expandafter

```

```

\protected@xdef\csname glo@\glo@label @useriaccess\endcsname{%
  \@glo@useriaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @useriiiaccess\endcsname{%
  \@glo@useriiiaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @userivaccess\endcsname{%
  \@glo@userivaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @uservaccess\endcsname{%
  \@glo@uservaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @userviaccess\endcsname{%
  \@glo@userviaccess}%
}
\newcommand*\glentryaccess}[1]{%
  \@gls@entry@field{#1}{access}%
}
\newcommand*\glentrytextaccess}[1]{%
  \@gls@entry@field{#1}{textaccess}%
}
\newcommand*\glentryfirstaccess}[1]{%
  \@gls@entry@field{#1}{firstaccess}%
}
\newcommand*\glentrypluralaccess}[1]{%
  \@gls@entry@field{#1}{pluralaccess}%
}
\newcommand*\glentryfirstpluralaccess}[1]{%
  \@gls@entry@field{#1}{firstpluralaccess}%
}
\newcommand*\glentrysymbolaccess}[1]{%
  \@gls@entry@field{#1}{symbolaccess}%
}
\newcommand*\glentrysymbolpluralaccess}[1]{%
  \@gls@entry@field{#1}{symbolpluralaccess}%
}
\newcommand*\glentrydescaccess}[1]{%
  \@gls@entry@field{#1}{descaccess}%
}
\newcommand*\glentrydescpluralaccess}[1]{%
  \@gls@entry@field{#1}{descpluralaccess}%
}
\newcommand*\glentryshortaccess}[1]{%
  \@gls@entry@field{#1}{shortaccess}%
}
\newcommand*\glentryshortpluralaccess}[1]{%
  \@gls@entry@field{#1}{shortpluralaccess}%
}
\newcommand*\glentrylongaccess}[1]{%
  \@gls@entry@field{#1}{longaccess}%
}

```

```

}
\newcommand*{\glsentrylongpluralaccess}[1]{%
  \@gls@entry@field{#1}{longpluralaccess}%
}
\newcommand*{\glsentryuseriaccess}[1]{%
  \@gls@entry@field{#1}{useriaccess}%
}
\newcommand*{\glsentryuseriiaccess}[1]{%
  \@gls@entry@field{#1}{useriiaccess}%
}
\newcommand*{\glsentryuseriiiaccess}[1]{%
  \@gls@entry@field{#1}{useriiiaccess}%
}
\newcommand*{\glsentryuserivaccess}[1]{%
  \@gls@entry@field{#1}{userivaccess}%
}
\newcommand*{\glsentryuservaccess}[1]{%
  \@gls@entry@field{#1}{uservaccess}%
}
\newcommand*{\glsentryuserviaccess}[1]{%
  \@gls@entry@field{#1}{userviaccess}%
}
}
\newcommand{\glsfieldaccsupp}[4]{%
  \ifdef\glscategory
  {%
    \ifcsdef{glsxtr\glscategory{#4}#3accsupp}%
    {\csname glsxtr\glscategory{#4}#3accsupp\endcsname{#1}{#2}}%
    {%
      \ifcsdef{glsxtr\glscategory{#4}accsupp}%
      {\csname glsxtr\glscategory{#4}accsupp\endcsname{#1}{#2}}%
      {%
        \ifcsdef{gls#3accsupp}%
        {\csname gls#3accsupp\endcsname{#1}{#2}}%
        {\glsaccsupp{#1}{#2}}%
      }%
    }%
  }%
  {%
    \ifcsdef{gls#3accsupp}%
    {\csname gls#3accsupp\endcsname{#1}{#2}}%
    {\glsaccsupp{#1}{#2}}%
  }%
}
\newcommand{\xglsfieldaccsupp}[1]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsfieldaccsupp\expandafter{\@gls@replacementtext}%
}
\newcommand*{\glsshortaccsupp}[2]{\glsaccessibility{E}{#1}{#2}}
\newcommand*{\glsshortplaccsupp}{\glsshortaccsupp}
\newcommand*{\glsaccsupp}[2]{\glsaccessibility{ActualText}{#1}{#2}}

```

```

\newcommand*\xglsaccsupp}[2]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsaccsupp\expandafter{\@gls@replacementtext}{#2}%
}
\newcommand*\@gls@access@display}[2]{%
  \protected@edef\@glo@access{#2}%
  \ifx\@glo@access\@gls@noaccess
    #1%
  \else
    \xglsaccsupp{\@glo@access}{#1}%
  \fi
}
\newcommand*\@gls@fieldaccess@display}[4]{%
  \protected@edef\@glo@access{#4}%
  \ifdefequal\@glo@access\@gls@noaccess
    {#3}%
  {\expandafter\glsfieldaccsupp\expandafter{\@glo@access}{#3}{#2}{#1}}%
}
\newrobustcmd*\@glsnameaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@access}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{name}{#1}{\glsentryaccess{#2}}}%
}
\newrobustcmd*\@gls@textaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@textaccess}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{text}{#1}{\glsentrytextaccess{#2}}}%
}
\newrobustcmd*\@gls@pluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@pluralaccess}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{plural}{#1}{\glsentrypluralaccess{#2}}}%
}
\newrobustcmd*\@gls@firstaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@firstaccess}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{first}{#1}{\glsentryfirstaccess{#2}}}%
}
\newrobustcmd*\@gls@firstpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@firstpluralaccess}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{firstpl}{#1}{\glsentryfirstpluralaccess{#2}}}%
}
\newrobustcmd*\@gls@symbolaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@symbolaccess}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{symbol}{#1}{\glsentrysymbolaccess{#2}}}%
}
\newrobustcmd*\@gls@symbolpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@symbolpluralaccess}%

```



```

    {#1}%
    {\@gls@fieldaccess@display{#2}{symbolplural}{#1}{\glsentrysymbolpluralaccess{#2}}}%
  }
\newrobustcmd*{\glsdescriptionaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#2}@descaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{desc}{#1}{\glsentrydescaccess{#2}}}%
}
\newrobustcmd*{\glsdescriptionpluralaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#2}@descpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{descplural}{#1}{\glsentrydescpluralaccess{#2}}}%
}
\newrobustcmd*{\glsshortaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#2}@shortaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{short}{#1}{\glsentryshortaccess{#2}}}%
}
\newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#2}@shortpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glsentryshortpluralaccess{#2}}}%
}
\newrobustcmd*{\glslongaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#2}@longaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{long}{#1}{\glsentrylongaccess{#2}}}%
}
\newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#2}@longpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{longpl}{#1}{\glsentrylongpluralaccess{#2}}}%
}
\newrobustcmd*{\glsuseriaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#2}@useriaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useri}{#1}{\glsentryuseriaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#2}@useriiaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{userii}{#1}{\glsentryuseriiaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiiaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#2}@useriiiaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useriii}{#1}{\glsentryuseriiiaccess{#2}}}%
}
\newrobustcmd*{\glsuserivaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#2}@userivaccess}%

```

```

    {#1}%
    {\@gls@fieldaccess@display{#2}{useriv}{#1}{\glsentryuserivaccess{#2}}}%
  }
\newrobustcmd*{\glsuservaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@uservaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{userv}{#1}{\glsentryuservaccess{#2}}}%
}
\newrobustcmd*{\glsuserviaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@userviaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{uservi}{#1}{\glsentryuserviaccess{#2}}}%
}
\newrobustcmd*{\glsaccessdisplay}[3]{%
  \ifcsundef{gls#1accessdisplay}%
  {%
    \PackageError{glossaries-accsupp}{No accessibility support
      for key ‘#1’}{%
    }%
    {%
      \csname gls#1accessdisplay\endcsname{#2}{#3}%
    }%
  }
}
\renewcommand*{\@@gls@default@entryfmt}[2]{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          #2{\glspluralaccessdisplay
            {\glsentryplural{\glslabel}}{\glslabel}}%
          {\glsdescriptionpluralaccessdisplay
            {\glsentrydescplural{\glslabel}}{\glslabel}}%
          {\glsymbolpluralaccessdisplay
            {\glsentrysymbolplural{\glslabel}}{\glslabel}}
          {\glsinsert}%
        }%
      }%
    }%
    #1{\glsfirstpluralaccessdisplay
      {\glsentryfirstplural{\glslabel}}{\glslabel}}%
    {\glsdescriptionpluralaccessdisplay
      {\glsentrydescplural{\glslabel}}{\glslabel}}%
    {\glsymbolpluralaccessdisplay
      {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
    {\glsinsert}%
  }%
}%
}

```

```

{%
  \ifglused\glslabel
  {%
    #2{\glspluralaccessdisplay
      {\Glsentryplural{\glslabel}}{\glslabel}}%
    {\glsdescriptionpluralaccessdisplay
      {\glsentrydescplural{\glslabel}}{\glslabel}}%
    {\glssymbolpluralaccessdisplay
      {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
  {%
    #1{\glsfirstpluralaccessdisplay
      {\Glsentryfirstplural{\glslabel}}{\glslabel}}%
    {\glsdescriptionpluralaccessdisplay
      {\glsentrydescplural{\glslabel}}{\glslabel}}%
    {\glssymbolpluralaccessdisplay
      {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
}%
{%
  \ifglused\glslabel
  {%
    \MakeUppercase{%
      #2{\glspluralaccessdisplay
        {\glsentryplural{\glslabel}}{\glslabel}}%
        {\glsdescriptionpluralaccessdisplay
          {\glsentrydescplural{\glslabel}}{\glslabel}}%
        {\glssymbolpluralaccessdisplay
          {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
        {\glsinsert}}%
    }%
    {%
      \MakeUppercase{%
        #1{\glsfirstpluralaccessdisplay
          {\glsentryfirstplural{\glslabel}}{\glslabel}}%
          {\glsdescriptionpluralaccessdisplay
            {\glsentrydescplural{\glslabel}}{\glslabel}}%
          {\glssymbolpluralaccessdisplay
            {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
          {\glsinsert}}%
        }%
      }%
    }%
  }%
  \glscapscase
  {%
    \ifglused\glslabel
    {%

```

```

#2{\glstextaccessdisplay
  {\glstentrytext{\glslabel}}{\glslabel}}%
{\glsdescriptionaccessdisplay
  {\glstentrydesc{\glslabel}}{\glslabel}}%
{\glssymbolaccessdisplay
  {\glstentrysymbol{\glslabel}}{\glslabel}}%
{\glsinsert}}%
}%
{%
#1{\glsfirstaccessdisplay
  {\glstentryfirst{\glslabel}}{\glslabel}}%
{\glsdescriptionaccessdisplay
  {\glstentrydesc{\glslabel}}{\glslabel}}%
{\glssymbolaccessdisplay
  {\glstentrysymbol{\glslabel}}{\glslabel}}%
{\glsinsert}}%
}%
}%
{%
\ifglsused\glslabel
{%
#2{\glstextaccessdisplay
  {\Glsentrytext{\glslabel}}{\glslabel}}%
{\glsdescriptionaccessdisplay
  {\Glsentrydesc{\glslabel}}{\glslabel}}%
{\glssymbolaccessdisplay
  {\Glsentrysymbol{\glslabel}}{\glslabel}}%
{\glsinsert}}%
}%
{%
#1{\glsfirstaccessdisplay
  {\Glsentryfirst{\glslabel}}{\glslabel}}%
{\glsdescriptionaccessdisplay
  {\Glsentrydesc{\glslabel}}{\glslabel}}%
{\glssymbolaccessdisplay
  {\Glsentrysymbol{\glslabel}}{\glslabel}}%
{\glsinsert}}%
}%
}%
{%
\ifglsused\glslabel
{%
\MakeUppercase{%
#2{\glstextaccessdisplay
  {\glstentrytext{\glslabel}}{\glslabel}}%
{\glsdescriptionaccessdisplay
  {\glstentrydesc{\glslabel}}{\glslabel}}%
{\glssymbolaccessdisplay
  {\glstentrysymbol{\glslabel}}{\glslabel}}%
{\glsinsert}}%

```

```

}%
{%
  \MakeUppercase{%
    #1{\glsfirstaccessdisplay
      {\glsentryfirst{\glslabel}}{\glslabel}}%
      {\glsdescriptionaccessdisplay
        {\glsentrydesc{\glslabel}}{\glslabel}}%
      {\glsymbolaccessdisplay
        {\glsentrysymbol{\glslabel}}{\glslabel}}%
      {\glsinsert}}%
  }%
}%
}%
}%
{%
  \ifglsused{\glslabel}%
  {%
    #2{\glscustomtext}%
    {\glsdescriptionaccessdisplay
      {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
      {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
  {%
    #1{\glscustomtext}%
    {\glsdescriptionaccessdisplay
      {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
      {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
}%
}
\renewcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          \glspluralaccessdisplay
            {\glsentryplural{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glsfirstpluralaccessdisplay
            {\glsentryfirstplural{\glslabel}}{\glslabel}%

```

```

    \glsinsert
  }%
}%
{%
  \ifglsused\glslabel
  {%
    \glspluralaccessdisplay
      {\Glsentryplural{\glslabel}}{\glslabel}%
    \glsinsert
  }%
  {%
    \glsfirstpluralaccessdisplay
      {\Glsentryfirstplural{\glslabel}}{\glslabel}%
    \glsinsert
  }%
}%
{%
  \ifglsused\glslabel
  {%
    \glspluralaccessdisplay
      {\mfirstucMakeUppercase{\glsentryplural{\glslabel}}}%
      {\glslabel}%
    \mfirstucMakeUppercase{\glsinsert}%
  }%
  {%
    \glsfirstpluralaccessdisplay
      {\mfirstucMakeUppercase{\glsentryfirstplural{\glslabel}}}%
      {\glslabel}%
    \mfirstucMakeUppercase{\glsinsert}%
  }%
}%
}%
{%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      \glstextaccessdisplay{\glsentrytext{\glslabel}}{\glslabel}%
      \glsinsert
    }%
    {%
      \glsfirstaccessdisplay{\glsentryfirst{\glslabel}}{\glslabel}%
      \glsinsert
    }%
  }%
}%
{%
  \ifglsused\glslabel
  {%
    \glstextaccessdisplay{\Glsentrytext{\glslabel}}{\glslabel}%
    \glsinsert
  }%
}

```

```

    }%
    {%
      \glsfirstaccessdisplay{\Glsentryfirst{\glslabel}}{\glslabel}%
      \glsinsert
    }%
  }%
  {%
    \ifglsused\glslabel
    {%
      \glstextaccessdisplay
      {\mfirstucMakeUppercase{\glsentrytext{\glslabel}}}{\glslabel}%
      \mfirstucMakeUppercase{\glsinsert}%
    }%
    {%
      \glsfirstaccessdisplay
      {\mfirstucMakeUppercase{\glsentryfirst{\glslabel}}}{\glslabel}%
      \mfirstucMakeUppercase{\glsinsert}%
    }%
  }%
}
}
}
}
\renewcommand*{\glsgenacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglsused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \acronymfont
          {\glsshortpluralaccessdisplay
            {\glsentryshortpl{\glslabel}}{\glslabel}}%
          \glsinsert
        }%
        {%
          \acronymfont
          {\glsshortpluralaccessdisplay
            {\Glsentryshortpl{\glslabel}}{\glslabel}}%
          \glsinsert
        }%
        {%
          \mfirstucMakeUppercase
          {\acronymfont
            {\glsshortpluralaccessdisplay
              {\glsentryshortpl{\glslabel}}{\glslabel}}%
          }%

```

```

        \glsinsert}%
    }%
}%
{%
    \glscapscase
    {%
        \acronymfont
        {\glsshortaccessdisplay{\glsentryshort{\glslabel}}{\glslabel}}%
        \glsinsert
    }%
    {%
        \acronymfont
        {\glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
        \glsinsert
    }%
    {%
        \mfirstucMakeUppercase
        {\acronymfont{%
            \glsshortaccessdisplay{\glsentryshort{\glslabel}}{\glslabel}}%
        \glsinsert}%
    }%
}%
}%
{%
    \glsifplural
    {%
        \glscapscase
        {%
            \genplacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \Genplacrfullformat{\glslabel}{\glsinsert}%
        }%
        \mfirstucMakeUppercase
        {\genplacrfullformat{\glslabel}{\glsinsert}}%
    }%
}%
{%
    \glscapscase
    {%
        \genacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
        \Genacrfullformat{\glslabel}{\glsinsert}%
    }%
    \mfirstucMakeUppercase
    {\genacrfullformat{\glslabel}{\glsinsert}}%
}%

```



```

    }%
  }%
}%
{%
  \glscustomtext
}%
}
\renewcommand*{\genacrfullformat}[2]{%
  \glslongaccessdisplay{\glsentrylong{#1}}{#1}#2\space
  (\glsshortaccessdisplay{\protect\firstacronymfont{\glsentryshort{#1}}}{#1})%
}
\renewcommand*{\Genacrfullformat}[2]{%
  \glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space
  (\glsshortaccessdisplay{\protect\firstacronymfont{\Glsentryshort{#1}}}{#1})%
}
\renewcommand*{\genplacrfullformat}[2]{%
  \glslongpluralaccessdisplay{\glsentrylongpl{#1}}{#1}#2\space
  (\glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
}
\renewcommand*{\Genplacrfullformat}[2]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}#2\space
  (\glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\Glsentryshortpl{#1}}}{#1})%
}
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\glsentryshort{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@Acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glsapsaps\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
}

```

```

}%
\glspostlinkhook
}
\def\@ACRshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glsshortaccessdisplay
{\MakeUppercase{\glentryshort{#2}}}{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@acrlong#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glslongaccessdisplay{\glentrylong{#2}}{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@Acrlong#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glslongaccessdisplay{\Glsentrylong{#2}}{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@ACRlong#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax

```

```

\let\glsifplural\@secondoftwo
\let\glsifcaps\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \acronymfont{\glslongaccessdisplay{%
    \MakeUppercase{\glsentrylong{#2}}}{#2}#3}%
  }%
  \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@gls@text@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@text@accessdisplay{\glsentrytext{#2}}}{#2}#3}%
}
\def\@Gls@text@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Gls@text@accessdisplay{\Glsentrytext{#2}}}{#2}#3}%
}
\def\@GLS@text@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\gls@text@accessdisplay{\mfirstucMakeUppercase{\glsentrytext{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}}%
}
}
\def\@gls@first@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@first@accessdisplay{\glsentryfirst{#2}}}{#2}#3}%
}
\def\@Gls@first@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Gls@first@accessdisplay{\Glsentryfirst{#2}}}{#2}#3}%
}
\def\@GLS@first@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\gls@first@accessdisplay{\mfirstucMakeUppercase{\glsentryfirst{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}}%
}
}
\def\@gls@plural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@plural@accessdisplay{\glsentryplural{#2}}}{#2}#3}%
}
\def\@Gls@plural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Gls@plural@accessdisplay{\Glsentryplural{#2}}}{#2}#3}%
}
\def\@GLS@plural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\gls@plural@accessdisplay{\mfirstucMakeUppercase{\glsentryplural{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}}%
}
}
\def\@gls@firstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@firstplural@accessdisplay{\glsentryfirstplural{#2}}}{#2}#3}%
}
\def\@Gls@firstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Gls@firstplural@accessdisplay{\Glsentryfirstplural{#2}}}{#2}#3}%
}
}

```

```

\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsfirstpluralaccessdisplay{\mfirstucMakeUppercase{\glsentryfirstplural{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glsentryname{#2}}{#2}#3}%
}
\def\@Glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\Glsentryname{#2}}{#2}#3}%
}
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsnameaccessdisplay{\mfirstucMakeUppercase{\glsentryname{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\glsentrydesc{#2}}{#2}#3}%
}
\def\@Glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
}
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsdescriptionaccessdisplay{\mfirstucMakeUppercase{\glsentrydesc{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\glsentrydescplural{#2}}{#2}#3}%
}
\def\@Glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
}
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsdescriptionpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrydescplural{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
}
\def\@Glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}
\def\@GLSsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsymbolaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbol{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsymbolplural@#1#2[#3]{%

```

```

    \@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\glstentrysymbolplural{#2}}{#2}#3}%
}
\def\@Glsymbolplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\glstentrysymbolplural{#2}}{#2}#3}%
}
\def\@GLSsymbolplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{%
        {\glssymbolpluralaccessdisplay{\mfirstucMakeUppercase{\glstentrysymbolplural{#2}}}{#2}%
        \mfirstucMakeUppercase{#3}}%
    }
}
\def\@glsuseri@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glstentryuseri{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
}
\def\@GLSuseri@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{%
        {\glsuseriaccessdisplay{\mfirstucMakeUppercase{\glstentryuseri{#2}}}{#2}%
        \mfirstucMakeUppercase{#3}}%
    }
}
\def\@glsuserii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glstentryuserii{#2}}{#2}#3}%
}
\def\@Glsuser@ii#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glsentryuserii{#2}}{#2}#3}%
}
\def\@GLSuserii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{%
        {\glsuseriiaccessdisplay{\mfirstucMakeUppercase{\glstentryuserii{#2}}}{#2}%
        \mfirstucMakeUppercase{#3}}%
    }
}
\def\@glsuseriii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\glstentryuseriii{#2}}{#2}#3}%
}
\def\@Glsuser@iii#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glsentryuseriii{#2}}{#2}#3}%
}
\def\@GLSuseriii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{%
        {\glsuseriiiaccessdisplay{\mfirstucMakeUppercase{\glstentryuseriii{#2}}}{#2}%
        \mfirstucMakeUppercase{#3}}%
    }
}
\def\@glsuseriv@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glstentryuseriv{#2}}{#2}#3}%
}
\def\@Glsuser@iv#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
}
\def\@GLSuseriv@#1#2[#3]{%

```

```

\@gls@field@link{#1}{#2}%
  {\glsuserivaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriv{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsuser@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glsentryuseriv{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
}
\def\@GLSuser@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuserivaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriv{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glsentryuseri{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuseriaccessdisplay{\mfirstucMakeUppercase{\glsentryuseri{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\glsentryname{#1}}{#1}}%
  }%
}
\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\Glsentryname{#1}}{#1}}%
  }%
}
\renewcommand*{\glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\glsentrydesc{#1}}{#1}%
  }%
}
\renewcommand*{\Glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}

```

```

\renewcommand*\glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\glsentrysymbol{#1}}{#1}%
  }%
}
\renewcommand*\Glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
  }%
}
\newcommand*\accsuppglossaryentryfield}[5]{%
  \glossaryentryfield{#1}%
  {\glsnameaccessdisplay{#2}{#1}}%
  {\glsdescriptionaccessdisplay{#3}{#1}}%
  {\glssymbolaccessdisplay{#4}{#1}}{#5}%
}
\newcommand*\accsuppglossarysubentryfield}[6]{%
  \glossarysubentryfield{#1}{#2}%
  {\glsnameaccessdisplay{#3}{#2}}%
  {\glsdescriptionaccessdisplay{#4}{#2}}%
  {\glssymbolaccessdisplay{#5}{#2}}{#6}%
}
\renewacronymstyle{long-short}%
{%
  \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
  \renewcommand*\genacrfullformat}[2]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}##2\space
    (\glsshortaccessdisplay
      {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
  }%
  \renewcommand*\Genacrfullformat}[2]{%
    \glslongaccessdisplay{\Glsentrylong{##1}}{##1}##2\space
    (\glsshortaccessdisplay
      {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
  }%
  \renewcommand*\genplacrfullformat}[2]{%
    \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}##2\space
    (\glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
  }%
  \renewcommand*\Genplacrfullformat}[2]{%
    \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}##2\space
    (\glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
  }%
}

```

```

\renewcommand*\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\renewacronymstyle{short-long}%
{%
  \ifglschaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat}[2]{%
  \glsshortaccessdisplay
  {\protect\firstacronymfont{\glentryshort{##1}}}{##1}##2\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*\Genacrfullformat}[2]{%
  \glsshortaccessdisplay
  {\protect\firstacronymfont{\Glsentryshort{##1}}}{##1}##2\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*\genplacrfullformat}[2]{%
  \glsshortpluralaccessdisplay
  {\protect\firstacronymfont{\glentryshortpl{##1}}}{##1}##2\space
  (\glslongpluralaccessdisplay
  {\glentrylongpl{##1}}{##1})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
  \glsshortpluralaccessdisplay
  {\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2\space
  (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})%
}%
\renewcommand*\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\renewacronymstyle{long-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%

```



```

        \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
    }
\renewacronymstyle{long-sc-short-desc}%
{%
    \GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
    \GlsUseAcrStyleDefs{long-sc-short}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymsort}[2]{##2}%
    \renewcommand*{\acronymentry}[1]{%
        \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
    }
\renewacronymstyle{long-sm-short-desc}%
{%
    \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
    \GlsUseAcrStyleDefs{long-sm-short}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymsort}[2]{##2}%
    \renewcommand*{\acronymentry}[1]{%
        \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
    }
\renewacronymstyle{short-long-desc}%
{%
    \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
    \GlsUseAcrStyleDefs{short-long}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymsort}[2]{##2}%
    \renewcommand*{\acronymentry}[1]{%
        \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
    }
\renewacronymstyle{sc-short-long-desc}%
{%
    \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
    \GlsUseAcrStyleDefs{sc-short-long}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymsort}[2]{##2}%
    \renewcommand*{\acronymentry}[1]{%
        \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
    }

```

```

}
\renewacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glslongpluralaccessdisplay{\glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay{\Glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay
            {\mfirstucMakeUppercase{\glsentrylongpl{\glslabel}}}{\glslabel}%
            {\mfirstucMakeUppercase{\glsinsert}}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \glslongaccessdisplay{\glsentrylong{\glslabel}}{\glslabel}\glsinsert
      }%
      {%
        \glslongaccessdisplay{\Glsentrylong{\glslabel}}{\glslabel}\glsinsert
      }%
      {%
        \glslongaccessdisplay
          {\mfirstucMakeUppercase
            {\glsentrylong{\glslabel}\glsinsert}}{\glslabel}%
          {\mfirstucMakeUppercase{\glsinsert}}%
      }%
    }%
  }%
}

```

```

    }%
  }%
  {%
    \glsgenentryfmt
  }%
}%
{\glscustomtext\glsinsert}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay
    {\mfirstucMakeUppercase{\glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\Glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\mfirstucMakeUppercase{\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\glsentryfull}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\glsentryfullpl}[1]{%

```

```

\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}\space
(\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*\Glsentryfullpl[1]{%
\glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space
(\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*\acronymentry[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}%
\renewcommand*\acronymsort[2]{##1}%
\renewcommand*\acronymfont[1]{##1}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\renewacronymstyle{dua-desc}%
{%
\GlsUseAcrEntryDispStyle{dua}%
}%
{%
\GlsUseAcrStyleDefs{dua}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymentry[1]{%
\glslongaccessdisplay{\acronymfont{\glsentrylong{##1}}}{##1}%
\renewcommand*\acronymsort[2]{##2}%
}%
\renewacronymstyle{footnote}%
{%
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*\genacrfullformat[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}{##1}}%
}%
\renewcommand*\Genacrfullformat[2]{%
\glsshortaccessdisplay
{\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}{##1}}%
}%
\renewcommand*\genplacrfullformat[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}}%
}%
\renewcommand*\Genplacrfullformat[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}}%
}

```

```

}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##2}}{##2}}##3\space
    (\glslongaccessdisplay{\glentrylong{##2}}{##2})}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##2}}{##2}}##3\space
    (\glslongaccessdisplay{\glentrylong{##2}}{##2})}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortaccessdisplay
      {\mfirstucMakeUppercase
        {\acronymfont{\glentryshort{##2}}{##2}}##3\space
        (\glslongaccessdisplay{\glentrylong{##2}}{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\glentryshortpl{##2}}{##2}}##3\space
      (\glslongpluralaccessdisplay{\glentrylongpl{##2}}{##2})}}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\glentryshortpl{##2}}{##2}}##3\space
      (\glslongpluralaccessdisplay{\glentrylongpl{##2}}{##2})}}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\mfirstucMakeUppercase
        {\acronymfont{\glentryshortpl{##2}}{##2}}##3\space
        (\glslongpluralaccessdisplay{\glentrylongpl{##2}}{##2})}}}%
\renewcommand*{\glentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}{##1}}\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1})}%
\renewcommand*{\Glentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}{##1}}\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1})}%
\renewcommand*{\glentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\glentryshortpl{##1}}{##1}}\space
    (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})}%
\renewcommand*{\Glentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\glentryshortpl{##1}}{##1}}\space
    (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})}%

```

```

}
\renewacronymstyle{footnote-sc}%
{
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glstextup{\glspluralsuffix}}%
}%
\renewacronymstyle{footnote-sm}%
{
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}%
\renewacronymstyle{footnote-desc}%
{
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sc-desc}%
{
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sm-desc}%
{
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}

```

```

}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
  }
}
\newcommand*{\glsdefaultshortaccess}[2]{#1}
\renewcommand*{\newacronymhook}{%
  \protected@edef\@gls@keylist{%
    shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    shortpluralaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    \the\glskeylisttok}%
  \expandafter\glskeylisttok\expandafter{\@gls@keylist}%
}
\renewcommand*{\DefaultNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      description={\the\glslongtok},%
      descriptionaccess=\relax,%
      text={\the\glsshorttok},%
      access={\noexpand\@glo@textaccess},%
      sort={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      first={\noexpand\glslongaccessdisplay
        {\the\glslongtok}{\the\glslabeltok}\space
        (\noexpand\glsshortaccessdisplay
          {\the\glsshorttok}{\the\glslabeltok})},%
      plural={\the\glsshorttok\acrpluralsuffix},%
      firstplural={\noexpand\glslongpluralaccessdisplay
        {\noexpand\@glo@longpl}{\the\glslabeltok}\space
        (\noexpand\glsshortpluralaccessdisplay
          {\noexpand\@glo@shortpl}{\the\glslabeltok})},%
      firstaccess=\relax,%
      firstpluralaccess=\relax,%
      textaccess={\noexpand\@glo@shortaccess},%
      \the\glskeylisttok
    }%
  }%
}
\let\@org@gls@assign@firstpl\gls@assign@firstpl

```

```

\let\org@gl@assign@plural\gl@assign@plural
\let\org@gl@assign@descplural\gl@assign@descplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@gl@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\org@gl@assign@firstpl
\let\gl@assign@plural\org@gl@assign@plural
\let\gl@assign@symbolplural\org@gl@assign@symbolplural
\@gl@noexpand@assign@default@plurals
}
\renewcommand*{\DescriptionFootnoteNewAcronymDef}{%
\edef\do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\gl@labeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\gl@shorttok}},%
    sort={\the\gl@shorttok},%
    text={\the\gl@shorttok},%
    short={\the\gl@shorttok},%
    shortplural={\the\gl@shorttok\noexpand\acrpluralsuffix},%
    shortaccess={\gl@defaultshortaccess{\the\gl@longtok}{\the\gl@shorttok}},%
    long={\the\gl@longtok},%
    longplural={\the\gl@longtok\noexpand\acrpluralsuffix},%
    access={\noexpand\@glo@textaccess},%
    plural={\the\gl@shorttok\noexpand\acrpluralsuffix},%
    symbol={\the\gl@longtok},%
    symbolplural={\the\gl@longtok\noexpand\acrpluralsuffix},%
    firstpluralaccess=\relax,
    textaccess={\noexpand\@glo@shortaccess},%
    \the\gl@keylisttok
  }%
}%
\let\org@gl@assign@firstpl\gl@assign@firstpl
\let\org@gl@assign@plural\gl@assign@plural
\let\org@gl@assign@symbolplural\gl@assign@symbolplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@symbolplural##1##2{%
  \@gl@expand@field{##1}{symbolplural}{##2}%
}

```



```

}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\DescriptionNewAcronymDef}{%
\edef\do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand
\acronymformat{\the\glsshorttok}{\the\glslongtok}},%
access={\noexpand\@glo@textaccess},%
sort={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
first={\the\glslongtok},%
firstaccess=\relax,
firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
text={\the\glsshorttok},%
textaccess={\the\glslongtok},%
plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
symbol={\noexpand\@glo@text},%
symbolaccess={\noexpand\@glo@textaccess},%
symbolplural={\noexpand\@glo@plural},%
firstpluralaccess=\relax,
textaccess={\noexpand\@glo@shortaccess},%
\the\glskeylisttok}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
\@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
\@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
\@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}

```

```

\renewcommand*{\FootnoteNewAcronymDef}{%
\edef\do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
text={\the\glsshorttok},%
textaccess={\the\glslongtok},%
access={\noexpand\@glo@textaccess},%
plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\the\glslongtok},%
descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
\@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
\@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
\@gls@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@descplural\@org@gls@assign@descplural
}
\renewcommand*{\SmallNewAcronymDef}{%
\edef\do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
access={\noexpand\@glo@symbolaccess},%
sort={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%

```

```

text={\noexpand\@glo@short},%
textaccess={\noexpand\@glo@shortaccess},%
plural={\noexpand\@glo@shortpl},%
first={\the\glslongtok},%
firstaccess=\relax,
firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\noexpand\@glo@first},%
descriptionplural={\noexpand\@glo@firstplural},%
symbol={\the\glsshorttok},%
symbolaccess={\the\glslongtok},%
symbolplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@@gls@expand@field{##1}{descplural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@descplural\@org@gls@assign@descplural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\glsshortaccesskey}{\glsshortkey access}%
\newcommand*{\glsshortpluralaccesskey}{\glsshortpluralkey access}%
\newcommand*{\glslongaccesskey}{\glslongkey access}%
\newcommand*{\glslongpluralaccesskey}{\glslongpluralkey access}%
\newcommand*{\showglongnameaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@access\endcsname
}
\newcommand*{\showglotextaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@textaccess\endcsname
}
\newcommand*{\showglopluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@pluralaccess\endcsname
}
\newcommand*{\showglofirstaccess}[1]{%

```

```

\expandafter\show\csname glo@glstetoklabel{#1}@firstaccess\endcsname
}
\newcommand*\showglofirstpluralaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@firstpluralaccess\endcsname
}
\newcommand*\showglosymbolaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@symbolaccess\endcsname
}
\newcommand*\showglosymbolpluralaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@symbolpluralaccess\endcsname
}
\newcommand*\showglodescaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@descaccess\endcsname
}
\newcommand*\showglodescpluralaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@descpluralaccess\endcsname
}
\newcommand*\showgloshortaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@shortaccess\endcsname
}
\newcommand*\showgloshortpluralaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@shortpluralaccess\endcsname
}
\newcommand*\showglolongaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@longaccess\endcsname
}
\newcommand*\showglolongpluralaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@longpluralaccess\endcsname
}
}

```

8.20 Rollback v4.49 (glossaries-babel-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-babel}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
\ForEachTrackedDialect{\this@dialect}{%
\IfTrackedLanguageFileExists{\this@dialect}%
{glossaries-}% prefix
{.ldf}%
}%
\RequireGlossariesLang{\CurrentTrackedTag}%
}%
{%
\PackageWarningNoLine{glossaries}%
{No language module detected for ‘\this@dialect’.\MessageBreak

```



```

\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}
\ifglxindy
  \renewcommand*{\GlsAddXdyLocation}[2]{%
    \edef\@xdyuserlocationdefs{%
      \@xdyuserlocationdefs ^^J%
      (define-location-class \string"#1\string"^^J\space\space
        \space(#2))
    }%
    \edef\@xdyuserlocationnames{%
      \@xdyuserlocationnames^^J\space\space\space
      \string"#1\string"%
    }
  }
\fi
\renewcommand{\@do@wrglossary}[1]{%
\ifglxindy
  \expandafter\@glo@check@mkidxrangear\@glsnumberformat\@nil
  \def\@glo@range{}%
  \expandafter\if\@glo@prefix(\relax
    \def\@glo@range{:open-range}%
  \else
    \expandafter\if\@glo@prefix)\relax
    \def\@glo@range{:close-range}%
  \fi
  \fi
  \protected@edef\@glslocref{\theglentrycounter}%
  \@gls@checkmkidchars\@glslocref
  \glossary[\csname glo@#1@type\endcsname]{%
    (indexentry :tkey (\csname glo@#1@index\endcsname)
      :locref \string"\@glslocref\string" %
      :attr \string"\@glo@suffix\string" \@glo@range
    )
  }%
\else
  \@set@glo@numformat\@glo@numfmt\@gls@counter\@glsnumberformat
  \glossary[\csname glo@#1@type\endcsname]{%
    \string\glossaryentry{\csname glo@#1@index\endcsname
      \@gls@encapchar\@glo@numfmt}{\theglentrycounter}}%
  \fi
}
\def\@set@glo@numformat#1#2#3{%
  \expandafter\@glo@check@mkidxrangear#3\@nil
  \protected@edef#1{%
    \@glo@prefix setentrycounter[]#{#2}%
  \expandafter\string\csname\@glo@suffix\endcsname
  }%
  \@gls@checkmkidchars#1%
}

```

```

\ifglxindy
\def\writeist{%
  \openout\glswrite=\istfilename
  \write\glswrite{;; xindy style file created by the glossaries
    package in compatible-2.07 mode}%
  \write\glswrite{;; for document '\jobname' on
    \the\year-\the\month-\the\day}%
  \write\glswrite{^^J; required styles^^J}
  \@for\@xdystyle:=\@xdyrequiredstyles\do{%
    \ifx\@xdystyle\@empty
    \else
      \protected@write\glswrite{{(require
        \string"\@xdystyle.xdy\string")}}%
    \fi
  }%
  \write\glswrite{^^J%
    ; list of allowed attributes (number formats)^^J}%
  \write\glswrite{(define-attributes ((\@xdyattributes)))}%
  \write\glswrite{^^J; user defined alphabets^^J}%
  \write\glswrite{\@xdyuseralphabets}%
  \write\glswrite{^^J; location class definitions^^J}%
  \protected@edef\@gls@roman{\@roman{0}\string"
    \string"roman-numbers-lowercase\string" :sep \string"}}%
  \@onelevel@sanitize\@gls@roman
  \edef\@tmp{\string" \string"roman-numbers-lowercase\string"
    :sep \string"}%
  \@onelevel@sanitize\@tmp
  \ifx\@tmp\@gls@roman
    \write\glswrite{(define-location-class
      \string"roman-page-numbers\string"^^J\space\space\space
      (\string"roman-numbers-lowercase\string")
      :min-range-length \@glsminrange)}%
  \else
    \write\glswrite{(define-location-class
      \string"roman-page-numbers\string"^^J\space\space\space
      (:sep "\@gls@roman")
      :min-range-length \@glsminrange)}%
  \fi
  \write\glswrite{(define-location-class
    \string"Roman-page-numbers\string"^^J\space\space\space
    (\string"roman-numbers-uppercase\string")
    :min-range-length \@glsminrange)}%
  \write\glswrite{(define-location-class
    \string"arabic-page-numbers\string"^^J\space\space\space
    (\string"arabic-numbers\string")
    :min-range-length \@glsminrange)}%
  \write\glswrite{(define-location-class
    \string"alpha-page-numbers\string"^^J\space\space\space
    (\string"alpha\string")
    :min-range-length \@glsminrange)}%

```

```

\write\glswrite{(define-location-class
  \string"Alpha-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"Appendix-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string"
  :sep \string"\@glsAlphacompositor\string"
  \string"arabic-numbers\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"arabic-section-numbers\string"^^J\space\space\space
  (\string"arabic-numbers\string"
  :sep \string"\glscompositor\string"
  \string"arabic-numbers\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\glsseeformat\string"
  :close \string"{}\string")}%
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
  :open \string"\string
  \glossarysection[\string\glossarytoctitle]{\string
  \glossarytitle}\string\glossary preamble\string~n\string\begin
  {theglossary}\string\glossaryheader\string~n\string" ^^J\space
  \space\space:close \string"\expandafter@gobble
  \string%\string~n\string
  \end{theglossary}\string\glossarypostamble
  \string~n\string" ^^J\space\space\space
  :tree)}}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\string~n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \string~n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\gls closebrace\string")}%
\write\glswrite{(markup-locref-list

```



```

:sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
:sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\string~n\string")}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
:open-head \string"\string\glsgroupheading
\glsopenbrace\string"^^J\space\space\space
:close-head \string"\glsclosebrace\string")}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%
\noist}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}
\def\writeist{\relax
\openout\glswrite=\istfilename
\write\glswrite{\expandafter\@gobble\string\% makeindex style file
created by the glossaries package}
\write\glswrite{\expandafter\@gobble\string\% for document
'\jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\@gls@actualchar'}
\write\glswrite{encap '\@gls@encapchar'}
\write\glswrite{level '\@gls@levelchar'}
\write\glswrite{quote '\@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
\glossarytoctitle]{\string\glossarytitle}\string
\glossarypreamble\string\n\string\begin{theglossary}\string
\glossaryheader\string\n\string"}

```

```

\write\glswrite{postamble \string"\string%\string\n\string
  \end{theglossary}\string\glossarypostamble\string\n
  \string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
  \string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}
\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
  \string"\string\relax \string\glsresetentrylist\string\n
  \string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
  \string"\string\relax \string\glsresetentrylist\string\n
  \string"}
\write\glswrite{delim_0 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_2 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_t \string"\string\}\string}\string"}
\write\glswrite{delim_n \string"\string\delimN \string"}
\write\glswrite{delim_r \string"\string\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
  \string"\string}\string\relax
  \string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"glscompositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF@empty
\else
  \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\noist
}
\fi
\renewcommand*{\noist}{\let\writeist\relax}

```

8.22 Rollback v4.49 (glossaries-compatible-307-2021-11-01.sty)

Version 4.49 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-307}[2021/11/01 v4.49 (NLCT)]
\newcommand{\compatglossarystyle}[2]{%
  \ifcsundef{@glscompstyle@#1}%
  {%
    \csdef{@glscompstyle@#1}{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary compatibility style ‘#1’ is already defined}{}%
  }%
}
\compatglossarystyle{inline}{%
  \renewcommand{\glossaryentryfield}[5]{%
    \glsinlinedopostchild
    \gls@inlinesep
    \def\glo@desc{##3}%
    \def\@no@post@desc{\nopostdesc}%
    \glsentryitem{##1}\glsinlinenameformat{##1}{##2}%
    \ifx\glo@desc\@no@post@desc
      \glsinlineemptydescformat{##4}{##5}%
    \else
      \ifstrempy{##3}%
        {\glsinlineemptydescformat{##4}{##5}}%
        {\glsinlinedescformat{##3}{##4}{##5}}%
      \fi
    \ifglshaschildren{##1}%
    {%
      \glsresetsubentrycounter
      \glsinlineparentchildseparator
      \def\gls@inlinesubsep{}%
      \def\gls@inlinepostchild{\glsinlinepostchild}%
    }%
    {}%
    \def\gls@inlinesep{\glsinlineseparator}%
  }%
  \renewcommand{\glossarysubentryfield}[6]{%
    \gls@inlinesubsep%
    \glsinlinesubnameformat{##2}{##3}%
    \glssubentryitem{##2}\glsinlinesubdescformat{##4}{##5}{##6}%
    \def\gls@inlinesubsep{\glsinlinesubseparator}%
  }%
}
\compatglossarystyle{list}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \item[\glsentryitem{##1}\glstarget{##1}{##2}]
```

```

        ##3\glspostdescription\space ##5}%
    \renewcommand*\glossarysubentryfield}[6]{%
        \glsentryitem{##2}%
        \glstarget{##2}{\strut}##4\glspostdescription\space ##6.}%
    }
\compatglossarystyle{listgroup}{%
    \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{listhypergroup}{%
    \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{altlist}{%
    \renewcommand*\glossaryentryfield}[5]{%
        \item[\glsentryitem{##1}\glstarget{##1}{##2}]%
        \mbox{}\par\nobreak\@afterheading
        ##3\glspostdescription\space ##5}%
    \renewcommand*\glossarysubentryfield}[6]{%
        \par
        \glsentryitem{##2}%
        \glstarget{##2}{\strut}##4\glspostdescription\space ##6}%
    }%
\compatglossarystyle{altlistgroup}{%
    \csuse{@glscompstyle@altlist}%
}%
\compatglossarystyle{altlisthypergroup}{%
    \csuse{@glscompstyle@altlist}%
}%
\compatglossarystyle{listdotted}{%
    \renewcommand*\glossaryentryfield}[5]{%
        \item[]\makebox[\glslistdottedwidth][l]{%
            \glsentryitem{##1}\glstarget{##1}{##2}%
            \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##3}%
    \renewcommand*\glossarysubentryfield}[6]{%
        \item[]\makebox[\glslistdottedwidth][l]{%
            \glsentryitem{##2}%
            \glstarget{##2}{##3}%
            \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##4}%
    }%
\compatglossarystyle{sublistdotted}{%
    \csuse{@glscompstyle@listdotted}%
    \renewcommand*\glossaryentryfield}[5]{%
        \item[\glsentryitem{##1}\glstarget{##1}{##2}]%
    }%
\compatglossarystyle{long}{%
    \renewcommand*\glossaryentryfield}[5]{%
        \glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
    \renewcommand*\glossarysubentryfield}[6]{%
        &
        \glsentryitem{##2}%
        \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
    }%

```

```

}%
\compatglossarystyle{longborder}{%
\csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheader}{%
\csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheaderborder}{%
\csuse{@glscompstyle@long}%
}%
\compatglossarystyle{long3col}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4 & ##6\\}%
}%
\compatglossarystyle{long3colborder}{%
\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long3colheader}{%
\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long3colheaderborder}{%
\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long4col}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
}%
\compatglossarystyle{long4colheader}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{long4colborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{long4colheaderborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4col}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colheader}{%
\csuse{@glscompstyle@long4col}%
}%

```

```

\compatglossarystyle{altlong4colborder}{%
  \csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colheaderborder}{%
  \csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{longragged}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glstarget{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
    \tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6%
    \tabularnewline}%
}%
\compatglossarystyle{longraggedborder}{%
  \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longraggedheader}{%
  \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longraggedheaderborder}{%
  \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longragged3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glstarget{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{longragged3colborder}{%
  \csuse{@glscompstyle@longragged3col}%
}%
\compatglossarystyle{longragged3colheader}{%
  \csuse{@glscompstyle@longragged3col}%
}%
\compatglossarystyle{longragged3colheaderborder}{%
  \csuse{@glscompstyle@longragged3col}%
}%
\compatglossarystyle{altlongragged4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glstarget{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%

```

```

\compatglossarystyle{altlongragged4colheader}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colborder}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colheaderborder}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{index}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item\glstarget{##1}{##2}%
    \ifx\relax##4\relax
      \else
        \space{##4}%
      \fi
    \space ##3\glspostdescription \space ##5}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    \ifcase##1\relax
      % level 0
      \item
    \or
      % level 1
      \subitem
      \glssubentryitem{##2}%
    \else
      % all other levels
      \subsubitem
    \fi
    \textbf{\glstarget{##2}{##3}}%
    \ifx\relax##5\relax
      \else
        \space{##5}%
      \fi
    \space##4\glspostdescription\space ##6}%
}%
\compatglossarystyle{indexgroup}{%
  \csuse{@glscompstyle@index}%
}%
\compatglossarystyle{indexhypergroup}{%
  \csuse{@glscompstyle@index}%
}%
\compatglossarystyle{tree}{%
  \renewcommand{\glossaryentryfield}[5]{%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstarget{##1}{##2}%
    \ifx\relax##4\relax
      \else
        \space{##4}%
      \fi
    \fi
  }%

```

```

\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\textbf{\glstarget{##2}{##3}}%
\ifx\relax##5\relax
\else
\space(##5)%
\fi
\space##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treegroup}{%
\csuse{@glscmpstyle@tree}%
}%
\compatglossarystyle{treehypergroup}{%
\csuse{@glscmpstyle@tree}%
}%
\compatglossarystyle{treenoname}{%
\renewcommand{\glossaryentryfield}[5]{%
\hangindent0pt\relax
\parindent0pt\relax
\glssubentryitem{##1}\textbf{\glstarget{##1}{##2}}%
\ifx\relax##4\relax
\else
\space(##4)%
\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\glstarget{##2}{\strut}%
##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treenonamegroup}{%
\csuse{@glscmpstyle@treenoname}%
}%
\compatglossarystyle{treenonamehypergroup}{%
\csuse{@glscmpstyle@treenoname}%
}%
\compatglossarystyle{almtree}{%
\renewcommand{\glossaryentryfield}[5]{%
\ifnum\@gls@prevlevel=0\relax
\else

```



```

        \settowidth{\glstreeindent}{\textbf{\@glswidestname\space}}%
        \hangindent\glstreeindent
        \parindent\glstreeindent
    \fi
    \makebox[0pt][r]{\makebox[\glstreeindent][l]{%
        \glstentryitem{##1}\textbf{\glstarget{##1}{##2}}}%
    \ifx\relax##4\relax
    \else
        (##4)\space
    \fi
    ##3\glspostdescription\space##5\par
    \def\@gls@prevlevel{0}%
}%
\renewcommand{\glossarysubentryfield}[6]{%
    \ifnum##1=1\relax
        \glssubentryitem{##2}%
    \fi
    \ifnum\@gls@prevlevel=##1\relax
    \else
        \@ifundefined{@glswidestname\romannumeral##1}{%
            \settowidth{\gls@tmplen}{\textbf{\@glswidestname\space}}{%
            \settowidth{\gls@tmplen}{\textbf{%
                \csname @glswidestname\romannumeral##1\endcsname\space}}}%
        \ifnum\@gls@prevlevel<##1\relax
            \setlength\glstreeindent\gls@tmplen
            \addtolength\glstreeindent\parindent
            \parindent\glstreeindent
        \else
            \@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
                \settowidth{\glstreeindent}{\textbf{%
                    \@glswidestname\space}}}%
                \settowidth{\glstreeindent}{\textbf{%
                    \csname @glswidestname\romannumeral\@gls@prevlevel
                    \endcsname\space}}}%
            \addtolength\parindent{-\glstreeindent}%
            \setlength\glstreeindent\parindent
        \fi
    \fi
    \hangindent\glstreeindent
    \makebox[0pt][r]{\makebox[\gls@tmplen][l]{%
        \textbf{\glstarget{##2}{##3}}}%
    \ifx##5\relax\relax
    \else
        (##5)\space
    \fi
    ##4\glspostdescription\space##6\par
    \def\@gls@prevlevel{##1}%
}%
}%
\compatglossarystyle{almtreeegroup}{%

```

```

\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{almtreehypergroup}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolindex}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexgroup}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexhypergroup}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcoltree}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenoname}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenonamegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenonamehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolalmtree}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolalmtreegroup}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolalmtreehypergroup}{%
\csuse{@glscompstyle@almtree}%
}%
}%

\compatglossarystyle{superragged}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4\glspostdescription\space ##6%
}

```

```

        \tabularnewline}%
}%
\compatglossarystyle{superraggedborder}{%
  \csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheader}{%
  \csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheaderborder}{%
  \csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superragged3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{superragged3colborder}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheader}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheaderborder}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{altsuperragged4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%
\compatglossarystyle{altsuperragged4colheader}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colborder}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colheaderborder}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{super}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &

```

```

        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}##4\glspostdescription\space ##6\}%
}%
\compatglossarystyle{superborder}{%
  \csuse{@glscompstyle@super}%
}%
\compatglossarystyle{superheader}{%
  \csuse{@glscompstyle@super}%
}%
\compatglossarystyle{superheaderborder}{%
  \csuse{@glscompstyle@super}%
}%
\compatglossarystyle{super3col}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glssubentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\}%
}%
\compatglossarystyle{super3colborder}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super3colheader}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super3colheaderborder}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super4col}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glssubentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\}%
}%
\compatglossarystyle{super4colheader}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{super4colborder}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{super4colheaderborder}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4col}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colheader}{%

```

```

\csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colborder}{%
\csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colheaderborder}{%
\csuse{@glscompstyle@super4col}%
}%

```

8.23 Rollback v4.49 (glossaries-polyglossia-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-polyglossia}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
\ForEachTrackedDialect{\this@dialect}{%
\IfTrackedLanguageFileExists{\this@dialect}%
{glossaries-}% prefix
{.ldf}%
{%
\RequireGlossariesLang{\CurrentTrackedTag}%
}%
}%
\PackageWarningNoLine{glossaries}%
{No language module detected for ‘\this@dialect’.\MessageBreak
Language modules need to be installed separately.\MessageBreak
Please check on CTAN for a bundle called\MessageBreak
‘glossaries-\CurrentTrackedLanguage’ or similar}%
}%
}%
}%
{}%

```

8.24 Rollback v4.49 (glossaries-prefix-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-prefix}[2021/11/01 v4.49 (NLCT)]
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
\ProcessOptions
\RequirePackage{glossaries}[=v4.49]
\define@key{glossentry}{prefixfirst}{\def\@glo@entryprefixfirst{#1}}%
\define@key{glossentry}{prefixfirstplural}{\def\@glo@entryprefixfirstplural{#1}}%
\define@key{glossentry}{prefix}{\def\@glo@entryprefix{#1}}%
\define@key{glossentry}{prefixplural}{\def\@glo@entryprefixplural{#1}}%
\appto{@gls@keymap}{,%

```

```

{prefixfirst}{prefixfirst},%
{prefixfirstplural}{prefixfirstplural},%
{prefix}{prefix},%
{prefixplural}{prefixplural}%
}
\appto@newglossaryentryprehook{%
\def@glo@entryprefix{}%
\def@glo@entryprefixplural{}%
\let@glo@entryprefixfirst@gls@default@value
\let@glo@entryprefixfirstplural@gls@default@value
}
\appto@newglossaryentryposthook{%
\gls@assign@field{}{\@glo@label}{prefix}{\@glo@entryprefix}%
\gls@assign@field{}{\@glo@label}{prefixplural}{\@glo@entryprefixplural}%
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @prefix\endcsname}{\@glo@label}{prefixfirst}%
{\@glo@entryprefixfirst}%
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @prefixplural\endcsname}{\@glo@label}%
{prefixfirstplural}{\@glo@entryprefixfirstplural}%
}
\newcommand*{\glsentryprefixfirst}[1]{\csuse{glo@glsdetoklabel{#1}@prefixfirst}}
\newcommand*{\glsentryprefixfirstplural}[1]{%
\csuse{glo@glsdetoklabel{#1}@prefixfirstplural}}
\newcommand*{\glsentryprefix}[1]{\csuse{glo@glsdetoklabel{#1}@prefix}}
\newcommand*{\glsentryprefixplural}[1]{\csuse{glo@glsdetoklabel{#1}@prefixplural}}
\newrobustcmd*{\Glsentryprefixfirst}[1]{%
\protected@edef@glo@text{\csname glo@glsdetoklabel{#1}@prefixfirst\endcsname}%
\xmakefirstuc@glo@text
}
\newrobustcmd*{\Glsentryprefixfirstplural}[1]{%
\protected@edef@glo@text{\csname glo@glsdetoklabel{#1}@prefixfirstplural\endcsname}%
\xmakefirstuc@glo@text
}
\newrobustcmd*{\Glsentryprefix}[1]{%
\protected@edef@glo@text{\csname glo@glsdetoklabel{#1}@prefix\endcsname}%
\xmakefirstuc@glo@text
}
\newrobustcmd*{\Glsentryprefixplural}[1]{%
\protected@edef@glo@text{\csname glo@glsdetoklabel{#1}@prefixplural\endcsname}%
\xmakefirstuc@glo@text
}
\newcommand*{\ifglshasprefix}[3]{%
\ifcsemtyp{glo@glsdetoklabel{#1}@prefix}%
{#3}%
{#2}%
}
\newcommand*{\ifglshasprefixplural}[3]{%
\ifcsemtyp{glo@glsdetoklabel{#1}@prefixplural}%
{#3}%
}

```

```

    {#2}%
}
\newcommand*{\ifglshasprefixfirst}[3]{%
  \ifcempty{glo@glstdetoklabel{#1}@prefixfirst}%
  {#3}%
  {#2}%
}
\newcommand*{\ifglshasprefixfirstplural}[3]{%
  \ifcempty{glo@glstdetoklabel{#1}@prefixfirstplural}%
  {#3}%
  {#2}%
}
\providecommand{\@glsprefix@record@hook}[2]{%
  \ifdef\@glsextr@record
  {\@glsextr@record{#1}{#2}{glslink}}%
  {}%
}
\newcommand{\glsprefixsep}{}
\newrobustcmd{\pgls}{\@glshyp@opt\@pgls}
\newcommand*{\@pgls}[2][{}]{%
  \new@ifnextchar[%
  {\@pgls@{#1}{#2}}%
  {\@pgls@{#1}{#2}[]}%
}
\def\@pgls@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glstdoifexists{#2}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefix{#2}{\glsentryprefix{#2}\glsprefixsep}{%
      }%
      {%
        \ifglshasprefixfirst{#2}{\glsentryprefixfirst{#2}\glsprefixsep}{%
        }%
        \@glss@{#1}{#2}[#3]%
      }%
    }%
  }
}
\newrobustcmd{\pglsp1}{\@glshyp@opt\@pglsp1}
\newcommand*{\@pglsp1}[2][{}]{%
  \new@ifnextchar[%
  {\@pglsp1@{#1}{#2}}%
  {\@pglsp1@{#1}{#2}[]}%
}
\def\@pglsp1@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glstdoifexists{#2}%
  {%
    \ifglused{#2}%
    {%

```

```

        \ifglshasprefixplural{#2}{\glstentryprefixplural{#2}\glsprefixsep}{}%
    }%
    {%
        \ifglshasprefixfirstplural{#2}%
            {\glstentryprefixfirstplural{#2}\glsprefixsep}{}%
    }%
    \@glspl@{#1}{#2}[#3]%
}%
}
\newrobustcmd{\Pglsh}{\@gls@hyp@opt\Pglsh}
\newcommand*{\@Pglsh}[2][{}]{%
    \new@ifnextchar[%
        {\@Pglsh@{#1}{#2}}%
        {\@Pglsh@{#1}{#2}[{}]}%
    }
\def\@Pglsh@#1#2[#3]{%
    \@glsprefix@record@hook{#1}{#2}%
    \glsdoifexists{#2}%
    {%
        \ifglsused{#2}%
        {%
            \ifglshasprefix{#2}%
            {%
                \Glsentryprefix{#2}%
                \glsprefixsep
                \@gls@{#1}{#2}[#3]%
            }%
            {\@Gls@{#1}{#2}[#3]}%
        }%
        {%
            \ifglshasprefixfirst{#2}%
            {%
                \Glsentryprefixfirst{#2}%
                \glsprefixsep
                \@gls@{#1}{#2}[#3]%
            }%
            {\@Gls@{#1}{#2}[#3]}%
        }%
    }%
}
\newrobustcmd{\Pglspl}{\@gls@hyp@opt\Pglspl}
\newcommand*{\@Pglspl}[2][{}]{%
    \new@ifnextchar[%
        {\@Pglspl@{#1}{#2}}%
        {\@Pglspl@{#1}{#2}[{}]}%
    }
\def\@Pglspl@#1#2[#3]{%
    \@glsprefix@record@hook{#1}{#2}%
    \glsdoifexists{#2}%
    {%

```



```

\ifglsused{#2}%
{%
  \ifglschasprefixplural{#2}%
  {%
    \Glsentryprefixplural{#2}%
    \glsprefixsep
    \@glspl@{#1}{#2}[#3]%
  }%
  {\@Glspl@{#1}{#2}[#3]}%
}%
{%
  \ifglschasprefixfirstplural{#2}%
  {%
    \Glsentryprefixfirstplural{#2}%
    \glsprefixsep
    \@glspl@{#1}{#2}[#3]%
  }%
  {\@Glspl@{#1}{#2}[#3]}%
}%
}%
}
\newrobustcmd{\PGLS}{\@gls@hyp@opt\PGLS}
\newcommand*{\@PGLS}[2][{}]{%
  \new@ifnextchar[{}]{\@PGLS@{#1}{#2}}%
  {\@PGLS@{#1}{#2}[{}]}%
}
\def\@PGLS@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglschasprefix{#2}%
      {\mfirstucMakeUppercase{\glsentryprefix{#2}\glsprefixsep}}}%
    }%
    {%
      \ifglschasprefixfirst{#2}%
      {\mfirstucMakeUppercase{\glsentryprefixfirst{#2}\glsprefixsep}}}%
    }%
    \@GLS@{#1}{#2}[#3]%
  }%
}
\newrobustcmd{\PGLSp1}{\@gls@hyp@opt\PGLSp1}
\newcommand*{\@PGLSp1}[2][{}]{%
  \new@ifnextchar[{}]{\@PGLSp1@{#1}{#2}}%
  {\@PGLSp1@{#1}{#2}[{}]}%
}
\def\@PGLSp1@#1#2[#3]{%

```

```

\@glsprefix@record@hook{#1}{#2}%
\glsdoifexists{#2}%
{%
  \ifglsused{#2}%
  {%
    \ifglschasprefixplural{#2}%
    {\mfirstucMakeUppercase{\glsentryprefixplural{#2}\glsprefixsep}}{}}%
  }%
  {%
    \ifglschasprefixfirstplural{#2}%
    {\mfirstucMakeUppercase{\glsentryprefixfirstplural{#2}\glsprefixsep}}{}}%
  }%
  \@GLSp1@{#1}{#2}{#3}%
}%
}

```

8.25 Rollback v4.49 (glossary-hypernav-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-hypernav}[2021/11/01 v4.49 (NLCT)]
\newcommand*\glsnavhyperlink}[3][\@glo@type]{%
  \protected@edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%
  \glslink{\glsnavhyperlinkname{#1}{#2}}{#3}}
\newcommand*\glsnavhyperlinkname}[2]{\glsn:#1@#2}
\newcommand*\glsnavhypertarget}[3][\@glo@type]{%
  \glsnavhypertarget{#1}{#2}{#3}%
}
\newcommand*\@glsnavhypertarget}[3]{%
  \protected@write\@auxout{\string\@gls@hypergroup{#1}{#2}}%
  \@gls@target{\glsnavhyperlinkname{#1}{#2}}{#3}%
  \expandafter\let
  \expandafter\@gls@list\csname @gls@hypergroup@#1\endcsname
  \@for\@gls@elem:=\@gls@list\do{%
    \ifthenelse{\equal{\@gls@elem}{#2}}{\@endfortrue}{}}%
  \if@endfor
  \else
  \GlossariesWarningNoLine{Navigation panel
    for glossary type ‘#1’^^Jmissing group ‘#2’}%
  \gdef\gls@hypergroup@rerun{%
    \GlossariesWarningNoLine{Navigation panel
      has changed. Rerun LaTeX}}%
  \fi
}
\let\gls@hypergroup@rerun\relax
\AtEndDocument{\gls@hypergroup@rerun}
\newcommand*\@gls@hypergroup}[2]{%
\@ifundefined{\@gls@hypergroup@list@#1}{%
  \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{#2}%
}{%

```

```

\expandafter\let\expandafter\@gls@tmp
  \csname @gls@hypergroupplist@#1\endcsname
\expandafter\xdef\csname @gls@hypergroupplist@#1\endcsname{%
  \@gls@tmp,#2}%
}%
}
\newcommand*{\glsnavigation}{%
  \def\@gls@between{}%
  \ifcsundef{@gls@hypergroupplist@\@glo@type}%
  {%
    \def\@gls@list{}%
  }%
  {%
    \expandafter\let\expandafter\@gls@list
      \csname @gls@hypergroupplist@\@glo@type\endcsname
  }%
  \@for\@gls@tmp:=\@gls@list\do{%
    \@gls@between
    \@gls@getgrouptitle{\@gls@tmp}{\@gls@grptitle}%
    \glsnavhyperlink{\@gls@tmp}{\@gls@grptitle}%
    \let\@gls@between\glshypernavsep
  }%
}
\newcommand*{\glshypernavsep}{\space\textbar\space}
\newcommand*{\glsymbolnav}{%
  \glsnavhyperlink{glsymbols}{\glsgetgrouptitle{glsymbols}}%
  \glshypernavsep
  \glsnavhyperlink{glsnumbers}{\glsgetgrouptitle{glsnumbers}}%
  \glshypernavsep
}

```

8.26 Rollback v4.49 (glossary-inline-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-inline}[2021/11/01 v4.49 (NLCT)]
\newglossarystyle{inline}{%
  \renewenvironment{theglossary}%
  {%
    \def\gls@inlinesep{}%
    \def\gls@inlinesubsep{}%
    \def\gls@inlinepostchild{}%
  }%
  {\glspostinline}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgrupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsinlinedopostchild
    \gls@inlinesep
    \glsentryitem{##1}%
  }%
}

```

```

\glsinlinenameformat{##1}{%
  \glossentryname{##1}%
}%
\ifglsdescsuppressed{##1}%
{%
  \glsinlineemptydescformat
  {%
    \glossentrysymbol{##1}%
  }%
  {%
    ##2%
  }%
}%
{%
  \ifglshasdesc{##1}%
  {\glsinlinedescformat{\glossentrydesc{##1}}{\glossentrysymbol{##1}}{##2}}%
  {\glsinlineemptydescformat{\glossentrysymbol{##1}}{##2}}%
}%
\ifglshaschildren{##1}%
{%
  \glsresetsubentrycounter
  \glsinlineparentchildseparator
  \def\gls@inlinesubsep{}%
  \def\gls@inlinepostchild{\glsinlinepostchild}%
}%
{}%
\def\gls@inlinesep{\glsinlineseparator}%
}%
\renewcommand{\subglossentry}[3]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{%
    \glossentryname{##2}}%
  \glssubentryitem{##2}%
  \glsinlinesubdescformat{\glossentrydesc{##2}}{\glossentrysymbol{##2}}{##3}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%
\renewcommand*{\glsgroupskip}{}%
}
\newcommand*{\glsinlinedopostchild}{%
  \gls@inlinepostchild
  \def\gls@inlinepostchild{}%
}
\newcommand*{\glsinlineseparator}{;\space}
\newcommand*{\glsinlinesubseparator}{,\space}
\newcommand*{\glsinlineparentchildseparator}{:\space}
\newcommand*{\glsinlinepostchild}{}
\newcommand*{\glspostinline}{\glspostdescription\space}
\newcommand*{\glsinlinenameformat}[2]{\glstarget{#1}{#2}}
\newcommand*{\glsinlinedescformat}[3]{\space#1}
\newcommand*{\glsinlineemptydescformat}[2]{}

```

```

\newcommand*\glsinlinesubnameformat}[2]{\glstarget{#1}{}}
\newcommand*\glsinlinesubdescformat}[3]{#1}

```

8.27 Rollback v4.49 (glossary-list-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossary-list}[2021/11/01 v4.49 (NLCT)]
\providecommand*\indexspace{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*\glslistgroupheaderfmt}[1]{#1}
\newcommand*\glslistnavigationitem}[1]{\item[#1]}
\newcommand*\glslistinit{%
  \ifdef\GetTitleStringDisableCommands
  {%
    \GetTitleStringSetup{expand}%
    \GetTitleStringDisableCommands{%
      \let\glsentryitem\@gobble
      \let\glstarget\@secondoftwo
      \let\glossentryname\glslistexpandedname
      \let\glslistgroupheaderfmt\@firstofone
      \let\glsgetgrouptitle\@firstofone
      \let\glsnavhypertarget\@secondoftwo
      \let\glsnavigation\relax
    }%
  }%
}
\newcommand{\glslistexpandedname}[1]{%
  \glsunexpandedfieldvalue{#1}{name}%
}
\newglossarystyle{list}{%
  \renewenvironment{theglossary}%
    {\glslistinit\begin{description}}{\end{description}}%
  \renewcommand*\{glossaryheader}{}%
  \renewcommand*\{glsgroupheading}[1]{}%
  \renewcommand*\{glossentry}[2]{%
    \item[\glsentryitem{##1}%
      \glstarget{##1}{\glossentryname{##1}}]
      \glossentrydesc{##1}\glspostdescription\space ##2}%
  \renewcommand*\{subglossentry}[3]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\space
    \glossentrydesc{##2}\glspostdescription\space ##3.}%
  \renewcommand*\{glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{listgroup}{%
  \setglossarystyle{list}%
}

```

```

\renewcommand*\glsgroupheading}[1]{%
  \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
\newglossarystyle{listhypergroup}{%
  \setglossarystyle{list}%
  \renewcommand*\glossaryheader{%
    \glslistnavigationitem{\glsnavigation}}%
  \renewcommand*\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}]}]}
\newglossarystyle{altlist}{%
  \setglossarystyle{list}%
  \renewcommand*\glossentry}[2]{%
    \item[\glsentryitem{##1}%
      \glstarget{##1}{\glossentryname{##1}}%
      \mbox{}\par\nobreak\@afterheading
      \glossentrydesc{##1}\glspostdescription\space ##2}%
  \renewcommand*\subglossentry}[3]{%
    \par
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space ##3}%
}
\newglossarystyle{altlistgroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
\newglossarystyle{altlisthypergroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*\glossaryheader{%
    \glslistnavigationitem{\glsnavigation}}%
  \renewcommand*\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}]}]}
\newglossarystyle{listdotted}{%
  \setglossarystyle{list}%
  \renewcommand*\glossentry}[2]{%
    \item[\makebox[\glslistdottedwidth][l]{%
      \glsentryitem{##1}%
      \glstarget{##1}{\glossentryname{##1}}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##1}}}%
  \renewcommand*\subglossentry}[3]{%
    \item[\makebox[\glslistdottedwidth][l]{%
      \glssubentryitem{##2}%
      \glstarget{##2}{\glossentryname{##2}}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##2}}}%
}
\newlength\glslistdottedwidth
\setlength{\glslistdottedwidth}{.5\hsize}
\newglossarystyle{sublistdotted}{%
  \setglossarystyle{listdotted}%
  \renewcommand*\glossentry}[2]{%

```

```

    \item[\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}}]{%
}

```

8.28 Rollback v4.49 (glossary-long-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-long}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{long}{%
  \renewenvironment{theglossary}{%
    {\begin{longtable}[lp{glsdescwidth}]}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
  \fi
}
\newglossarystyle{longborder}{%
  \setglossarystyle{long}%
  \renewenvironment{theglossary}{%
    \begin{longtable}[|lp{glsdescwidth}|]{\end{longtable}}%
    \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
  }
\newglossarystyle{longheader}{%
  \setglossarystyle{long}%
  \renewcommand*{\glossaryheader}{%
    \bfseries \entryname & \bfseries \descriptionname\tabularnewline\endhead}%
}

```

```

\newglossarystyle{longheaderborder}{%
  \setglossarystyle{longborder}%
  \renewcommand*{\glossaryheader}{%
    \hrule\bfseries \entryname & \bfseries
    \descriptionname\tabularnewline\hrule
    \endhead
    \hrule\endfoot}%
}
\newglossarystyle{long3col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsdescwidth}p{\glspagelistwidth}}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{long3colborder}{%
  \setglossarystyle{long3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hrule\endhead\hrule\endfoot}%
}
\newglossarystyle{long3colheader}{%
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long3colheaderborder}{%
  \setglossarystyle{long3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hrule
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hrule\endhead
    \hrule\endfoot}%
}

```



```

}
\newglossarystyle{long4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{long4colheader}{%
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long4colborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long4colheaderborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}
}

```

```

\newglossarystyle{altlong4col}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheader}{%
  \setglossarystyle{long4colheader}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colborder}{%
  \setglossarystyle{long4colborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheaderborder}{%
  \setglossarystyle{long4colheaderborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%
    {\end{longtable}}%
}
}

```

8.29 Rollback v4.49 (glossary-longbooktabs-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-longbooktabs}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{booktabs}
\RequirePackage{glossary-long}[=v4.46]
\RequirePackage{glossary-longragged}[=v4.46]
\newglossarystyle{long-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname & \bfseries
    \descriptionname\tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{long3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%

```

```

        \toprule \bfseries \entryname &
        \bfseries \descriptionname &
        \bfseries \pagelistname
        \tabularnewline\midrule\endhead
        \bottomrule\endfoot}%
\ifglsgroupskip
  \renewcommand*\{glsgroupskip}\{}%
\else
  \renewcommand*\{glsgroupskip}\{glspenaltygroupskip}%
\fi
}
\newglossarystyle{long4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long4col}%
  \renewcommand*\{glossaryheader}{%
    \toprule \bfseries \entryname &
    \bfseries \descriptionname &
    \bfseries \symbolname &
    \bfseries \pagelistname
    \tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*\{glsgroupskip}\{}%
  \else
    \renewcommand*\{glsgroupskip}\{glspenaltygroupskip}%
  \fi
}
\newglossarystyle{altlong4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long4col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}\lp{\glsdescwidth}\lp{\glspagelistwidth}}%
    {\end{longtable}}%
}
\newglossarystyle{longragged-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}\l>{\raggedright}p{\glsdescwidth}}%
    {\end{longtable}}%
}
\newglossarystyle{longragged3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}\l>{\raggedright}p{\glsdescwidth}%
     >{\raggedright}p{\glspagelistwidth}}%
    {\end{longtable}}%
}
\newglossarystyle{altlongragged4col-booktabs}{%

```

```

\glspatchLToutput
\setglossarystyle{altlong4col-booktabs}%
\renewenvironment{theglossary}%
  {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
    >{\raggedright}p{\glspagelistwidth}}}%
  {\end{longtable}}%
}
\newcommand*{\glsLTpenaltycheck}{%
\ifnum\outputpenalty=-50\vskip-\normalbaselineskip\relax\fi
}
\newcommand{\glspenaltygroupskip}{%
\noalign{\penalty-50\vskip\normalbaselineskip}}
\let\@gls@org@LT@output\LT@output
\newcommand*{\glsrestoreLToutput}{\let\LT@output\@gls@org@LT@output}
\newcommand*{\glspatchLToutput}{%
\renewcommand*{\LT@output}{%
\ifnum\outputpenalty <-\@Mi
\ifnum\outputpenalty > -\LT@end@pen
\LT@err{floats and marginpars not allowed in a longtable}\@ehc
\else
\setbox\z@\vbox{\unvbox\@cclv}%
\ifdim \ht\LT@lastfoot>\ht\LT@foot
\dimen@\pagegoal
\advance\dimen@-\ht\LT@lastfoot
\ifdim\dimen@<\ht\z@
\setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%
\@makecol
\@outputpage
\setbox\z@\vbox{\box\LT@head\glsLTpenaltycheck}%
\fi
\fi
\global\@colroom\@colht
\global\vsizel\@colht
{\unvbox\z@\box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%
\fi
\else
\setbox\@cclv\vbox{\unvbox\@cclv\copy\LT@foot\vss}%
\@makecol
\@outputpage
\global\vsizel\@colroom
\copy\LT@head
\glsLTpenaltycheck
\nobreak
\fi
}%
}

```

8.30 Rollback v4.49 (glossary-longragged-2021-11-01.sty)

Version 4.49 preserved for rollback.

```
\ProvidesPackage{glossary-longragged}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{array}
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{longragged}{%
  \renewenvironment{theglossary}{%
    {\begin{longtable}[1>{\raggedright}p{glsdescwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2%
    \tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}%
    \glspostdescription\space ##3%
    \tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
  \fi
}
\newglossarystyle{longraggedborder}{%
  \setglossarystyle{longragged}%
  \renewenvironment{theglossary}{%
    \begin{longtable}{|l|>{\raggedright}p{glsdescwidth}|}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{longraggedheader}{%
  \setglossarystyle{longragged}%
  \renewcommand*{\glossaryheader}{%
    \bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\endhead}%
}
```

```

}
\newglossarystyle{longraggedheaderborder}{%
  \setglossarystyle{longraggedborder}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\hline
    \endhead
    \hline\endfoot}%
}
\newglossarystyle{longragged3col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}[1>{\raggedright}p{\glsdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{longragged3colborder}{%
  \setglossarystyle{longragged3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{longragged3colheader}{%
  \setglossarystyle{longragged3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{longragged3colheaderborder}{%
  \setglossarystyle{longragged3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline

```

```

        \bfseries\entryname&\bfseries\descriptionname&
        \bfseries\pagelistname\tabularnewline\hline\endhead
        \hline\endfoot}%
    }
\newglossarystyle{altlongragged4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}[l]>\raggedright}p{\glsdescwidth}l%
    >\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{altlongragged4colheader}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}[l]>\raggedright}p{\glsdescwidth}l%
    >\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{altlongragged4colborder}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>\raggedright}p{\glsdescwidth}||l%
    >\raggedright}p{\glspagelistwidth}||}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{altlongragged4colheaderborder}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%

```

```

    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|}
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}%
\renewcommand*{\glossaryheader}{%
  \hline\bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\hline\endhead
  \hline\endfoot}%
}

```

8.31 Rollback v4.49 (glossary-mcols-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-mcols}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{multicol}
\RequirePackage{glossary-tree}[=v4.49]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glsmcols}{2}
\newglossarystyle{mcolindex}{%
  \setglossarystyle{index}%
  \renewenvironment{theglossary}{%
    {%
      \begin{multicols}{\glsmcols}
      \setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%
      \let\item\glstreeitem
      \let\subitem\glstreesubitem
      \let\subsubitem\glstreesubsubitem
    }%
    {\end{multicols}}%
  }
\newglossarystyle{mcolindexgroup}{%
  \setglossarystyle{mcolindex}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\indexspace}%
  }
\newglossarystyle{mcolindexhypergroup}{%
  \setglossarystyle{mcolindex}%
  \renewcommand*{\glossaryheader}{%
    \item\glstreenavigationfmt{\glsnavigation}\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
    \indexspace}%
  }
\newglossarystyle{mcolindexspannav}{%
  \setglossarystyle{index}%

```



```

\renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
    \let\item\glstreeitem}%
  {\end{multicols}}%
\renewcommand*{\glsgroupheading}[1]{%
  \item\glstreegroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
  \indexspace}%
}
\newglossarystyle{mcoltree}{%
  \setglossarystyle{tree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}
\newglossarystyle{mcoltreegroup}{%
  \setglossarystyle{mcoltree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreehypergroup}{%
  \setglossarystyle{mcoltree}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcoltreespannav}{%
  \setglossarystyle{tree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
}

```

```

\newglossarystyle{mcoltreenoname}{%
  \setglossarystyle{treenoname}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}
\newglossarystyle{mcoltreenonamegroup}{%
  \setglossarystyle{mcoltreenoname}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreenonamehypergroup}{%
  \setglossarystyle{mcoltreenoname}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcoltreenonamespannav}{%
  \setglossarystyle{treenoname}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcolalmtree}{%
  \setglossarystyle{almtree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
}
\newglossarystyle{mcolalmtreegroup}{%
  \setglossarystyle{mcolalmtree}%

```

```

\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcolalmtreehypergroup}{%
\setglossarystyle{mcolalmtree}%
\renewcommand*\glossaryheader{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcolalmtreespannav}{%
\setglossarystyle{almtree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsncols}[\noindent\glstreenavigationfmt{\glsnavigation}]
\def\@gls@prevlevel{-1}%
\mbox{}\par
}%
{\par\end{multicols}}%
\renewcommand*\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}

```

8.32 Rollback v4.49 (glossary-super-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-super}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
\newlength{glsdescwidth}
\setlength{glsdescwidth}{0.6\hsize}
}{}

```

```

\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{super}{%
  \renewenvironment{theglossary}%
    {\tablehead{}}\tabletail{}}%
    \begin{supertabular}{lp{glstdescwidth}}%
    {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgruppeheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}}%
    \begin{supertabular}{|lp{glstdescwidth}|}%
    {\end{supertabular}}%
}
\newglossarystyle{superheader}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries \entryname &
      \bfseries\descriptionname\tabularnewline}}%
    \tabletail{}%
    \begin{supertabular}{lp{glstdescwidth}}%
    {\end{supertabular}}%
}
\newglossarystyle{superheaderborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline\bfseries \entryname &
      \bfseries \descriptionname\tabularnewline\hline}}%
    \tabletail{\hline}
    \begin{supertabular}{|lp{glstdescwidth}|}%

```

```

        {\end{supertabular}}%
    }
\newglossarystyle{super3col}{%
  \renewenvironment{theglossary}%
    {\tablehead{}}\tabletail{}%
    \begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}%
    {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
  \fi
}
\newglossarystyle{super3colborder}{%
  \setglossarystyle{super3col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
    {\end{supertabular}}%
  }
\newglossarystyle{super3colheader}{%
  \setglossarystyle{super3col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\pagelistname\tabularnewline}\tabletail{}%
    \begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}%
    {\end{supertabular}}%
  }
\newglossarystyle{super3colheaderborder}{%
  \setglossarystyle{super3colborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname&\bfseries\descriptionname&
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
    {\end{supertabular}}%
  }
}

```

```

\newglossarystyle{super4col}{%
  \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}}%
    {\begin{supertabular}{1111}}{%
      \end{supertabular}}%
  \renewcommand*\glossaryheader{}%
  \renewcommand*\glsgroupheading}[1]{}%
  \renewcommand\glossentry[2]{%
    \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} & ##2\tabularnewline
  }%
  \renewcommand\subglossentry[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*\glsgroupskip{}%
  \else
    \renewcommand*\glsgroupskip}{& & \tabularnewline}%
  \fi
}
\newglossarystyle{super4colheader}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}%
    \tabletail{}}%
    {\begin{supertabular}{1111}}%
    {\end{supertabular}}%
}
\newglossarystyle{super4colborder}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}{|1|1|1|1|}}%
    {\end{supertabular}}%
}
\newglossarystyle{super4colheaderborder}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}{|1|1|1|1|}}%
    {\end{supertabular}}%
}

```

```

}
\newglossarystyle{altsuper4col}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{}}\tabletail{}%
    \begin{supertabular}{lp{\glsdescwidth}lp{\glspagelistwidth}}%
    {\end{supertabular}}%
}
\newglossarystyle{altsuper4colheader}{%
  \setglossarystyle{super4colheader}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}\tabletail{}%
    \begin{supertabular}{lp{\glsdescwidth}lp{\glspagelistwidth}}%
    {\end{supertabular}}%
}
\newglossarystyle{altsuper4colborder}{%
  \setglossarystyle{super4colborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}%
      {|lp{\glsdescwidth}|lp{\glspagelistwidth}|}%
    {\end{supertabular}}%
}
\newglossarystyle{altsuper4colheaderborder}{%
  \setglossarystyle{super4colheaderborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname &
      \bfseries\descriptionname &
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}%
      {|lp{\glsdescwidth}|lp{\glspagelistwidth}|}%
    {\end{supertabular}}%
}
}

```

8.33 Rollback v4.49 (glossary-superragged-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-superragged}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{array}
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{%
}

```

```

\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{superragged}{%
  \renewenvironment{theglossary}%
    {\tablehead{} \tabletail{}}%
    {\begin{supertabular}{1>{\raggedright}p{\glsdescwidth}}}%
    {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2%
    \tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3%
    \tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superraggedborder}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline} \tabletail{\hline}%
    \begin{supertabular}{||1>{\raggedright}p{\glsdescwidth}||}}%
    {\end{supertabular}}%
}
\newglossarystyle{superraggedheader}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries \entryname & \bfseries \descriptionname
    \tabularnewline}%
    \tabletail{}}%
    {\begin{supertabular}{1>{\raggedright}p{\glsdescwidth}}}%
    {\end{supertabular}}%
}
\newglossarystyle{superraggedheaderborder}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline\bfseries \entryname &
    \bfseries \descriptionname\tabularnewline\hline}%
}

```



```

        \tabletail{\hline}
        \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}||}%
        {\end{supertabular}}%
    }
\newglossarystyle{superragged3col}{%
\renewenvironment{theglossary}%
{\tablehead{\tabletail}{%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}%
>{\raggedright}p{\glspagelistwidth}}}%
{\end{supertabular}}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1} &
##2\tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2} &
##3\tabularnewline
}%
\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi
}
\newglossarystyle{superragged3colborder}{%
\setglossarystyle{superragged3col}%
\renewenvironment{theglossary}%
{\tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}||%
>{\raggedright}p{\glspagelistwidth}||}%
{\end{supertabular}}%
}
\newglossarystyle{superragged3colheader}{%
\setglossarystyle{superragged3col}%
\renewenvironment{theglossary}%
{\tablehead{\bfseries\entryname&\bfseries\descriptionname&
\bfseries\pagelistname\tabularnewline}\tabletail}{%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}%
>{\raggedright}p{\glspagelistwidth}}}%
{\end{supertabular}}%
}
\newglossarystyle{superragged3colheaderborder}{%
\setglossarystyle{superragged3colborder}%
\renewenvironment{theglossary}%
{\tablehead{\hline

```

```

        \bfseries\entryname&\bfseries\descriptionname&
        \bfseries\pagelistname\tabularnewline\hline}%
\tabletail{\hline}%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
>{\raggedright}p{\glspagelistwidth}|}%
\end{supertabular}}%
}
\newglossarystyle{altsuperragged4col}{%
\renewenvironment{theglossary}%
{\tablehead{}\tabletail{}%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
>{\raggedright}p{\glspagelistwidth}|}%
\end{supertabular}}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1} &
\glossentrysymbol{##1} & ##2\tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2} &
\glossentrysymbol{##2} & ##3\tabularnewline
}%
\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi
}
\newglossarystyle{altsuperragged4colheader}{%
\setglossarystyle{altsuperragged4col}%
\renewenvironment{theglossary}%
{\tablehead{\bfseries\entryname&\bfseries\descriptionname&
\bfseries\symbolname &
\bfseries\pagelistname\tabularnewline}\tabletail{}%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
>{\raggedright}p{\glspagelistwidth}|}%
\end{supertabular}}%
}
\newglossarystyle{altsuperragged4colborder}{%
\setglossarystyle{altsuper4col}%
\renewenvironment{theglossary}%
{\tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}%
{|l|>{\raggedright}p{\glsdescwidth}|l|%
>{\raggedright}p{\glspagelistwidth}|}%
\end{supertabular}}%
}

```

```

}
\newglossarystyle{altsuperragged4colheaderborder}{%
  \setglossarystyle{altsuperragged4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname &
      \bfseries\descriptionname &
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}%
      \l|>{\raggedright}p{\glsdescwidth}|l|%
      >{\raggedright}p{\glspagelistwidth}|}%
    {\end{supertabular}}%
}

```

8.34 Rollback v4.49 (glossary-tree-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-tree}[2021/11/01 v4.49 (NLCT)]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glstreenamefmt}[1]{\textbf{#1}}
\newcommand*{\glstreegroupheaderfmt}[1]{\glstreenamefmt{#1}}
\newcommand*{\glstreenavigationfmt}[1]{\glstreenamefmt{#1}}
\ifdef\@idxitem
{\newcommand{\glstreeitem}{\@idxitem}}
{\newcommand{\glstreeitem}{\par\hangindent40\p@}}
\ifdef\subitem
{\let\glstreesubitem\subitem}
{\newcommand\glstreesubitem{\glstreeitem\hspace*{20\p@}}}
\ifdef\subsubitem
{\let\glstreesubsubitem\subsubitem}
{\newcommand\glstreesubsubitem{\glstreeitem\hspace*{30\p@}}}
\newcommand{\glstreepredesc}{\space}
\newcommand{\glstreechildpredesc}{\space}
\newglossarystyle{index}{%
  \renewenvironment{theglossary}%
    {\setlength{\parindent}{0pt}%
     \setlength{\parskip}{0pt plus 0.3pt}%
     \let\item\glstreeitem
     \let\subitem\glstreesubitem
     \let\subsubitem\glstreesubsubitem
    }%
    {\par}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand*{\glossentry}[2]{%

```

```

        \item\glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
        \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
        \glstreepredesc \glossentrydesc{##1}\glspostdescription\space ##2%
    }%
\renewcommand{\subglossentry}[3]{%
    \ifcase##1\relax
        % level 0
        \item
    \or
        % level 1
        \subitem
        \glssubentryitem{##2}%
    \else
        % all other levels
        \subsubitem
    \fi
    \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
    \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
    \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3%
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
\newglossarystyle{indexgroup}{%
    \setglossarystyle{index}%
    \renewcommand*{\glsgroupheading}[1]{%
        \item\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
        \indexspace
    }%
}
\newglossarystyle{indexhypergroup}{%
    \setglossarystyle{index}%
    \renewcommand*{\glossaryheader}{%
        \item\glstreenavigationfmt{\glsnavigation}\indexspace}%
    \renewcommand*{\glsgroupheading}[1]{%
        \item\glstreegroupheaderfmt
            {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
        \indexspace}%
}
\newglossarystyle{tree}{%
    \renewenvironment{theglossary}%
        {\setlength{\parindent}{0pt}%
         \setlength{\parskip}{0pt plus 0.3pt}}%
        {}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
        \hangindent0pt\relax
        \parindent0pt\relax
        \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
        \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
        \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
    }%
}

```

```

}%
\renewcommand{\subglossentry}[3]{%
  \hangindent##1\glstreeindent\relax
  \parindent##1\glstreeindent\relax
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
  \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
  \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3\par
}%
\renewcommand*{\glsgroupskip}{\ifglsgnogroupskip\else\indexspace\fi}}
\newglossarystyle{treegroup}{%
  \setglossarystyle{tree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par
    \indexspace}%
}
\newglossarystyle{treehypergroup}{%
  \setglossarystyle{tree}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsgroupnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt
    {\glsgrouphypertarget{##1}{\glsgrouptitle{##1}}}\par
    \indexspace}%
}
\newlength\glstreeindent
\setlength{\glstreeindent}{10pt}
\newglossarystyle{treenoname}{%
  \renewenvironment{theglossary}%
    {\setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}}%
    {}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glssentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
    \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
    \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
  }%
  \renewcommand{\subglossentry}[3]{%
    \hangindent##1\glstreeindent\relax
    \parindent##1\glstreeindent\relax
    \ifnum##1=1\relax
      \glssubentryitem{##2}%
    \fi

```

```

\glstarget{##2}{\strut}%
\glossentrydesc{##2}\glspostdescription\space##3\par
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{treenonamegroup}{%
\setglossarystyle{treenoname}%
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt
{\glsgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{treenonamehypergroup}{%
\setglossarystyle{treenoname}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt
{\glsnavigationtarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}
\newrobustcmd*{\glswidesttoplevelname}[1][\@glo@types]{%
\dimen@=0pt\relax
\gls@tmplen=0pt\relax
\forallglossaries[#1]{\@gls@type}%
{%
\forallglsentries[\@gls@type]{\@glo@label}%
{%
\ifglshasparent{\@glo@label}%
}%
{%
\settowidth{\dimen@}%
{\glstreenamefmt{\glsentryname{\@glo@label}}}%
\ifdim\dimen@>\gls@tmplen
\gls@tmplen=\dimen@
\letcs{\@glswidestname}{glo\@glsdetoklabel{\@glo@label}@name}%
\fi
}%
}%
}
\newcommand*{\glsssetwidest}[2][0]{%
\expandafter\def\csname @glswidestname\romannumeral#1\endcsname{%
#2}%
}
\newcommand*{\@glswidestname}{}
\newcommand*{\glstreenamebox}[2]{%
\makebox[#1][l]{#2}%
}
\newglossarystyle{almtree}{%

```

```

\renewenvironment{theglossary}%
  {\def\@gls@prevlevel{-1}%
   \mbox{}\par}%
  {\par}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
  \ifnum\@gls@prevlevel=0\relax
  \else
    \settowidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%
  \fi
  \hangindent\glstreeindent
  \parindent\glstreeindent
  \makebox[Opt][r]{\glstreenamebox{\glstreeindent}{%
    \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}}}%
  \ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}%
  \glossentrydesc{##1}\glspostdescription \space ##2\par
  \def\@gls@prevlevel{0}%
}%
\renewcommand{\subglossentry}[3]{%
  \ifnum##1=1\relax
  \glssubentryitem{##2}%
  \fi
  \ifnum\@gls@prevlevel=##1\relax
  \else
    \ifundefined{@glswidestname\romannumeral##1}{%
      \settowidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}{%
        \settowidth{\gls@tmplen}{\glstreenamefmt{%
          \csname @glswidestname\romannumeral##1\endcsname\space}}}%
      \ifnum\@gls@prevlevel<##1\relax
        \setlength\glstreeindent\gls@tmplen
        \addtolength\glstreeindent\parindent
        \parindent\glstreeindent
      \else
        \ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
          \settowidth{\glstreeindent}{\glstreenamefmt{%
            \@glswidestname\space}}{%
              \settowidth{\glstreeindent}{\glstreenamefmt{%
                \csname @glswidestname\romannumeral\@gls@prevlevel
                  \endcsname\space}}}%
          \addtolength\parindent{-\glstreeindent}%
          \setlength\glstreeindent\parindent
        \fi
        \fi
      \hangindent\glstreeindent
      \makebox[Opt][r]{\glstreenamebox{\gls@tmplen}{%
        \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
        \ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
        \glossentrydesc{##2}\glspostdescription\space ##3\par
        \def\@gls@prevlevel{##1}%

```

```

}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{almtreegroup}{%
\setglossarystyle{almtree}%
\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
\par\indexspace}%
}
\newglossarystyle{almtreehypergroup}{%
\setglossarystyle{almtree}%
\renewcommand*{\glossaryheader}{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}}

```

8.35 Rollback v4.52 (glossaries-2022-11-03.sty)

```

\NeedsTeXFormat{LaTeX2e}
\DeclareRelease{v4.46}{2020-03-19}{glossaries-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-2021-11-01.sty}
\DeclareCurrentRelease{v4.52}{2022-11-03}
\ProvidesPackage{glossaries}[2022/11/03 v4.52 (NLCT)]
\RequirePackage{ifthen}
\RequirePackage{xkeyval}[2006/11/18]
\RequirePackage{mfirstuc}
\newcommand{\glsentencecase}[1]{\makefirstuc{#1}}
\ExplSyntaxOn
\newcommand{\glslowercase}[1]{\text_lowercase:n{#1}}
\newcommand{\glsuppercase}[1]{\text_uppercase:n{#1}}
\ifdef\MFUsentencecase
{%
\newcommand{\glsmfuexcl}[1]{\MFUexcl{#1}}
\newcommand{\glsmfublocker}[1]{\MFUblocker{#1}}
\newcommand{\glsmfuaddmap}[2]{\MFUaddmap{#1}{#2}}
}
{

```



```

\RequirePackage{textcase}
\providecommand{\MFUsentencecase}[1]{\text_titlecase_first:n{#1}}
\providecommand{\glsmfuexcl}[1]{
  \tl_if_in:NnF \l_text_case_exclude_arg_tl {#1}
  {
    \tl_put_right:Nn \l_text_case_exclude_arg_tl {#1}
  }
}
\renewcommand*{\mfirstucMakeUppercase}{\MakeTextUppercase}
\newcommand{\glsmfublocker}[1]{\glsmfuexcl{#1}}
\newcommand{\glsmfuaddmap}[2]{\glsmfuexcl{#1}\glsmfublocker{#2}}
}
\ExplSyntaxOff
\RequirePackage{xfor}
\RequirePackage{datatool-base}
\RequirePackage{amsgen}
\RequirePackage{etoolbox}
\newif\if@gls@docloaded
\@ifpackageloaded{doc}{%
  {%
    \@gls@docloadedtrue
  }%
  {%
    \@ifclassloaded{nlctdoc}{\@gls@docloadedtrue}{\@gls@docloadedfalse}%
  }
}
\if@gls@docloaded
  \let\glsorg@theglossary\theglossary
  \let\glsorg@endtheglossary\endtheglossary
  \let\glsorg@PrintChanges\PrintChanges
  \renewcommand{\PrintChanges}{%
    \begingroup
      \let\theglossary\glsorg@theglossary
      \let\endtheglossary\glsorg@endtheglossary
      \glsorg@PrintChanges
    \endgroup
  }
\fi
\newif\if@gls@debug
\define@choicekey{glossaries.sty}{debug}[\gls@debug@val\gls@debug@nr]{%
{true,false,showtargets,showaccsupp}[true]{%
  \ifcase\gls@debug@nr\relax
    % debug=true
    \@gls@debugtrue
    \renewcommand*{\GlossariesWarning}[1]{%
      \PackageWarning{glossaries}{##1}%
    }%
    \renewcommand*{\GlossariesWarningNoLine}[1]{%
      \PackageWarningNoLine{glossaries}{##1}%
    }%
  \let\@gls@showtarget\@gobble

```

```

\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\or
% debug=false
\@gls@debugfalse
\let\@glsshowtarget\@gobble
\let\@glsshowaccsupp\@gobblethree
\PackageInfo{glossaries}{debug mode OFF}%
\or
% debug=showtargets
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowtarget}{\@glsshowtarget}%
\or
% debug=showaccsupp
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
\fi
}
\newcommand*\glsshowtarget}[1]{%
\ifmmode
\glsshowtargetinner{#1}%
\else
\ifinner
\glsshowtargetinner{#1}%
\else
\glsshowtargetouter{#1}%
\fi
\fi
}
\newcommand*\glsshowtargetfonttext}[1]{%
\ifmmode \nfss@text{\glsshowtargetfont #1}\else {\glsshowtargetfont #1}\fi
}
\newcommand*\glsshowtargetinner}[1]{\glsshowtargetfonttext{[#1]}}
\newcommand*\glsshowtargetouter}[1]{%
\glsshowtargetsymbol\marginpar{\glsshowtargetsymbol\glsshowtargetfont #1}}
\newcommand*\glsshowtargetsymbol}{\tiny$\triangleright$}
\newcommand*\glsshowtargetfont}{\ttfamily\footnotesize}

```

```

\newcommand*{\@glsshowtarget}[1]{}
\newcommand*{\glsdoshowtarget}[2]{\@glsshowtarget{#1}#2}
\newrobustcmd*{\@@glsshowtarget}[1]{%
  \begingroup
  \protected@edef\@gls@tmp{#1}%
  \@onelevel@sanitize\@gls@tmp
  \expandafter\glsshowtarget\expandafter{\@gls@tmp}%
  \endgroup
}
\newcommand*{\@glsshowaccsupp}[3]{}
\newrobustcmd*{\glsshowaccsupp}[3]{%
  \ifstrempy{#1}%
  {\@@glsshowtarget{/#2 (#3)}}%
  {\@@glsshowtarget{/#2 (#3) [#1]}}%
}
\newcommand*{\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries\space (or move
  \string\newglossaryentry\space
  definitions into the preamble)}%
  {You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key. It may
  be that the 'see' key has been written to the .glsdefs
  file from the previous run, in which case you need to
  move your definitions
  to the preamble if you don't want to use
  \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoinde}x{%
  [\gls@seenoinde@val\gls@seenoinde@nr]{error,warn,ignore}{%
  \ifcase\gls@seenoinde@nr
  \renewcommand*{\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries}%
  {You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key}%
  }%
  \or
  \renewcommand*{\@gls@see@noindex}{%
  \GlossariesWarning{'\gls@xr@key' key ignored}%
  }%
  \or
  \renewcommand*{\@gls@see@noindex}{}%
  \fi
}

```

```

\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}%
  {\newcommand*\@@glossarysec{section}}%
  {\newcommand*\@@glossarysec{chapter}}
\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
subsection,subsubsection,paragraph,subparagraph}[section]{%
  \renewcommand*\@@glossarysec{#1}}
\newcommand*\@@glossarysecstar{*}%
\newcommand*\@@glossaryseclabel{}%
\newcommand*\glsautoprefix{}%
\define@choicekey{glossaries.sty}{numberedsection}%
[\gls@numberedsection@val\gls@numberedsection@nr]{%
false,nolabel,autolabel,nameref}[nolabel]{%
  \ifcase\gls@numberedsection@nr\relax
    \renewcommand*\@@glossarysecstar{*}%
    \renewcommand*\@@glossaryseclabel{}%
  \or
    \renewcommand*\@@glossarysecstar{}%
    \renewcommand*\@@glossaryseclabel{}%
  \or
    \renewcommand*\@@glossarysecstar{}%
    \renewcommand*\@@glossaryseclabel{%
      \label{\glsautoprefix@glo@type}}%
  \or
    \renewcommand*\@@glossarysecstar{*}%
    \renewcommand*\@@glossaryseclabel{%
      \protected@edef\@currentlabelname{\glossarytoctitle}%
      \label{\glsautoprefix@glo@type}}%
  \fi
}
\@ifpackageloaded{classicthesis}
{\newcommand*\@glossary@default@style{index}}
{\newcommand*\@glossary@default@style{list}}
\define@key{glossaries.sty}{style}{%
  \def\@glossary@default@style{#1}}
}
\newcommand*\@gls@declareoption}[2]{%
  \DeclareOptionX{#1}{#2}%
  \DeclareOption{#1}{#2}%
}
\newcommand*\glossaryentrynumbers}[1]{#1\gls@save@numberlist{#1}}
\@gls@declareoption{nonumberlist}{%
  \renewcommand*\glossaryentrynumbers}[1]{\gls@save@numberlist{#1}}%
}
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{}
\gls@savenumberlistfalse
\newcommand*\@glo@seeautonumberlist{}
\@gls@declareoption{seeautonumberlist}{%
  \renewcommand*\@glo@seeautonumberlist{}%
}

```

```

\def\@glo@prefix{\glsnextpages}%
}%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{}
\glsesclocationsfalse
\newcommand*\@gls@loadlong{\RequirePackage{glossary-long}}
\@gls@declareoption{no long}{\renewcommand*\@gls@loadlong{}}
\IfFileExists{supertabular.sty}{%
\newcommand*\@gls@loadsuper{\RequirePackage{glossary-super}}}{%
\newcommand*\@gls@loadsuper{}}
\@ifpackageloaded{xtab}{\renewcommand*\@gls@loadsuper{}}{}
\@gls@declareoption{nosuper}{\renewcommand*\@gls@loadsuper{}}
\newcommand*\@gls@loadlist{\RequirePackage{glossary-list}}
\@gls@declareoption{no list}{%
\renewcommand*\@gls@loadlist}{%
\ifdefstring{\@glossary@default@style}{list}%
{\let\@glossary@default@style\relax}%
}}%
}%
}
\newcommand*\@gls@loadtree{\RequirePackage{glossary-tree}}
\@gls@declareoption{notree}{\renewcommand*\@gls@loadtree{}}
\@gls@declareoption{nostyles}{%
\renewcommand*\@gls@loadlong}{%
\renewcommand*\@gls@loadsuper}{%
\renewcommand*\@gls@loadlist}{%
\renewcommand*\@gls@loadtree}{%
\let\@glossary@default@style\relax
}
\newcommand*\@gls@postdescription}{%
\ifglsnopostdot\else.\spacefactor\sfcode`. \fi
}
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{}
\glsnopostdotfalse
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{}
\glsnogroupskipfalse
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{}
\@ifclassloaded{memoir}
{%
\glsucmarktrue
}%
{%
\glsucmarkfalse
}
\newcommand*\@gls@define@glossaryentrycounter}{%
\ifglsentrycounter
\ifundef\c@glossaryentry
{%
\ifx\@gls@counterwithin\@empty
\newcounter{glossaryentry}%

```

```

        \else
            \newcounter{glossaryentry}[\@gls@counterwithin]%
        \fi
        \def\theHglossaryentry{\currentglossary.\theglossaryentry}%
    }%
    {}%
\fi
}
\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\glstentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
    \renewcommand*{\@gls@counterwithin}{#1}%
    \glstentrycountertrue
    \@gls@define@glossaryentrycounter
}
\newcommand*{\@gls@counterwithin}{}
\newcommand{\@gls@define@glossarysubentrycounter}{%
    \ifundef\c@glossarysubentry
    {%
        \ifglssubentrycounter
            \ifglstentrycounter
                \newcounter{glossarysubentry}[glossaryentry]%
            \else
                \newcounter{glossarysubentry}%
            \fi
        \fi
        \def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
    }%
    {}%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
\glssubentrycounterfalse
\newcommand{\glsindexingsetting}{\ifglsxindy xindy\else makeindex\fi}
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none,clear}{%
    \renewcommand*{\@glo@default@sorttype}{#1}%
    \csname @gls@setupsort@#1\endcsname
}
\newcommand*{\glsprestandardsort}[3]{%
    \glsdosanitizesort
}
\newcommand*{\@glo@check@sortallowed}[1]{}
\newcommand*{\@gls@setupsort@standard}{%
    \def\do@glo@storeentry{\@glo@storeentry}%
    \def\@gls@defsortcount##1{%
        \def\@gls@defsort##1##2{%
            \ifx\@glo@sort\@glsdefaultsort
                \let\@glo@sort\@glo@name
            \fi
            \let\glsdosanitizesort\@gls@sanitizesort
        }
    }
}

```

```

\glsprestandardsort{\@glo@sort}{##1}{##2}%
\expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\@gls@setupsort@standard
\newcommand*\glsortnumberfmt [1]{%
\ifnum#1<100000 0\fi
\ifnum#1<10000 0\fi
\ifnum#1<1000 0\fi
\ifnum#1<100 0\fi
\ifnum#1<10 0\fi
\number#1%
}
\newcommand*\@gls@setupsort@def}{%
\def\do@glo@storeentry{\@glo@storeentry}%
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\ifcsundef{glossary@##1@sortcount}%
{\@gls@defsortcount{##1}}%
}%
\expandafter\global\expandafter
\advance\csname glossary@##1@sortcount\endcsname by 1\relax
\expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
\expandafter\glsortnumberfmt
{\csname glossary@##1@sortcount\endcsname}}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*\@gls@setupsort@use}{%
\let\do@glo@storeentry\@gobble
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\expandafter\gdef\csname glo@##2@sort\endcsname{}%
}%
\def\@gls@setsort##1{%
\protected@edef\@glo@parent{\csname glo@##1@parent\endcsname}%
\ifx\@glo@parent\@empty
\else
\expandafter\@gls@setsort\expandafter{\@glo@parent}%
\fi
\protected@edef\@glo@type{\csname glo@##1@type\endcsname}%

```

```

\protected@edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
\ifx\@gls@tmp\@empty
\expandafter\global\expandafter
\advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
\expandafter\protected@xdef\csname glo@##1@sort\endcsname{%
\expandafter\glssortnumberfmt
{\csname glossary@\@glo@type @sortcount\endcsname}}%
\@glo@storeentry{##1}%
\fi
}%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*{\@gls@setupsort@none}{%
\def\glsindexingsetting{none}%
\def\do@glo@storeentry##1{%
\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
\expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort
}%
\def\@gls@setsort##1{%
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
{Option sort=none not allowed with \string##1}%
{(Use sort=def instead)}}%
}
\newcommand*{\@gls@setupsort@clear}{%
\def\glsindexingsetting{none}%
\def\do@glo@storeentry##1{%
\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{\csgdef{glo@##2@sort}{}}%
\def\@gls@setsort##1{%
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
{Option sort=clear not allowed with \string##1}%
{(Use sort=def instead)}}%
}
\newcommand*{\glsdefmain}{%
\if@gls@docloaded
\newglossary[glg2]{main}{gls2}{glo2}{\glossaryname}%
\else
\newglossary{main}{gls}{glo}{\glossaryname}%
\fi
\newcommand*{\gls@tr@set@main@toctitle}{%
\translatelet{\glossarytoctitle}{Glossary}%
}%
}
\newcommand*{\glsdefaulttype}{main}
\newcommand*{\acronymtype}{\glsdefaulttype}
\@gls@declareoption{nomain}{%
\let\glsdefaulttype\relax
\renewcommand*{\glsdefmain}{%
}

```



```

\define@boolkey{glossaries.sty}[gls]{acronym}[true]{%
  \ifglsacronym
    \renewcommand{\@gls@do@acronymsdef}{%
      \DeclareAcronymList{acronym}%
      \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
      \renewcommand*{\acronymtype}{acronym}%
      \newcommand*\gls@tr@set@acronym@toctitle}{%
        \translatelet{\glossarytoctitle}{Acronyms}%
      }%
    }%
  \else
    \let\@gls@do@acronymsdef\relax
  \fi
}
\AtBeginDocument{%
  \ifglsacronym
    \providecommand*\printacronyms[1][1]{%
      \printglossary[type=\acronymtype,#1]%
    }%
  \fi
}
\newcommand*\@gls@do@acronymsdef{}
\@gls@declareoption{acronyms}{%
  \glsacronymtrue
  \def\@gls@do@acronymsdef{%
    \DeclareAcronymList{acronym}%
    \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
    \renewcommand*{\acronymtype}{acronym}%
    \newcommand*\gls@tr@set@acronym@toctitle}{%
      \translatelet{\glossarytoctitle}{Acronyms}%
    }%
  }%
}
\newcommand*\@gls@acronymlists{}
\newcommand*\@addtoacronymlists[1]{%
  \ifx\@gls@acronymlists\@empty
    \protected@xdef\@gls@acronymlists{#1}%
  \else
    \protected@xdef\@gls@acronymlists{\@gls@acronymlists,#1}%
  \fi
}
\newcommand*\DeclareAcronymList[1]{%
  \glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}
\newcommand*\glsIfListOfAcronyms[1]{%
  \protected@edef\@do@gls@islistofacronyms{%
    \noexpand\@gls@islistofacronyms{#1}{\@gls@acronymlists}}%
  \@do@gls@islistofacronyms
}
\newcommand*\@gls@islistofacronyms[4]{%
  \def\gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%

```

```

\def\@gls@before{##1}\def\@gls@after{##2}}%
\gls@islistofacronyms,#2,#1,\@nil\end\gls@islistofacronyms
\ifx\@gls@after\@nnil
#4%
\else
#3%
\fi
}
\newif\if@gls@isacronymlist
\newcommand*\@gls@checkisacronymlist}[1]{%
\glsIfListOfAcronyms{#1}%
{\@gls@isacronymlisttrue}{\@gls@isacronymlistfalse}%
}
\newcommand*\@SetAcronymLists}[1]{%
\renewcommand*\@gls@acronymlists}{#1}%
}
\define@key{glossaries.sty}{acronymlists}{%
\DeclareAcronymList{#1}%
}
\newcommand*\@gls@counter}{page}
\define@key{glossaries.sty}{counter}{%
\renewcommand*\@gls@counter}{#1}%
}
\newcommand*\@gls@nohyperlist}{%
\newcommand*\@GlsDeclareNoHyperList}[1]{%
\ifdefempty\@gls@nohyperlist
{%
\renewcommand*\@gls@nohyperlist}{#1}%
}%
}%
\appto\@gls@nohyperlist{,#1}%
}%
}
\define@key{glossaries.sty}{nohypertypes}{%
\GlsDeclareNoHyperList{#1}%
}
\newcommand*\@GlossariesWarning}[1]{%
\PackageWarning{glossaries}{#1}%
}
\newcommand*\@GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{#1}%
}
\newcommand*\@glosortentrieswarning}{%
\typeout{Using TeX to sort glossary entries---this may
take a while}%
}
\@gls@declareoption{nowarn}{%
\if@gls@debug
\GlossariesWarning{Warnings can't be suppressed in debug mode}%
\else

```

```

\renewcommand*\GlossariesWarning}[1]{%
\renewcommand*\GlossariesWarningNoLine}[1]{}%
\renewcommand*\glosortentrieswarning}{}%
\renewcommand*\@gls@missinglang@warn}[2]{}%
\fi
}
\newcommand*\@gls@missinglang@warn}[2]{%
\PackageWarningNoLine{glossaries}%
{No language module detected for ‘#1’.\MessageBreak
Language modules need to be installed separately.\MessageBreak
Please check on CTAN for a bundle called\MessageBreak
‘glossaries-#2’ or similar}%
}
\@gls@declareoption{noLANGwarn}{%
\renewcommand*\@gls@missinglang@warn}[2]{}%
}
\newcommand*\@gls@warnonglossdefined}{%
\GlossariesWarning{Overriding \string\printglossary}%
}
\newcommand*\@gls@warnontheglossdefined}{%
\GlossariesWarning{Overriding ‘theglossary’ environment}%
}
\@gls@declareoption{noREDEFwarn}{%
\renewcommand*\@gls@warnonglossdefined}{}%
\renewcommand*\@gls@warnontheglossdefined}{}%
}
\newcommand*\glssetexpandfield}[1]{%
\csdef{gls@assign@#1@field}##1##2{%
\@gls@expand@field{##1}{#1}{##2}%
}%
}
\newcommand*\glssetnoexpandfield}[1]{%
\csdef{gls@assign@#1@field}##1##2{%
\@gls@noexpand@field{##1}{#1}{##2}%
}%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}
\glssetnoexpandfield{descplural}
\newcommand*\@gls@sanitizename}{%
\glssetnoexpandfield{name}
\newcommand*\@gls@sanitizesymbol}{%
\glssetnoexpandfield{symbol}
\glssetnoexpandfield{symbolplural}
\newcommand*\@gls@sanitizesort}{%
\ifgls@sanitizesort
\@gls@sanitizesort
\else
\@gls@nosanitizesort
\fi

```

```

}
\newcommand*{\@gls@sanitizesort{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@gls@nosanitizesort}{%}
\newcommand*{\@gls@noidx@sanitizesort{%
  \ifdefvoid\@glo@sort
  }%
  {%
  \expandafter\@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
  \def\@glo@sort{#1#2}%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@gls@noidx@nosanitizesort}{%
  \ifdefvoid\@glo@sort
  }%
  {%
  \expandafter\@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
  \bgroup
  \glsnoidxstripaccents
  \protected@xdef\@glo@sort{#1#2}%
  \egroup
  \let\@glo@sort\@glo@sort
}
\newcommand*\glsnoidxstripaccents{%
  \let\IeC\@firstofone
  \let\add@accent@\@secondoftwo
  \let\@text@composite@x\@secondoftwo
  \let\@tabacckludge\@secondoftwo
  \expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
  \let'\@firstofone
  \let'\@firstofone
  \let^\@firstofone
  \let"@@firstofone
  \letu\@firstofone
  \lett\@firstofone
  \letd\@firstofone
  \letr\@firstofone
  \let=\@firstofone
  \let.\@firstofone
  \let~\@firstofone

```

```

\let\v\@firstofone
\let\H\@firstofone
\let\c\@firstofone
\let\b\@firstofone
\let\a\@secondoftwo
\def\AE{AE}%
\def\ae{ae}%
\def\OE{OE}%
\def\oe{oe}%
\def\AA{AA}%
\def\aa{aa}%
\def\L{L}%
\def\l{l}%
\def\O{O}%
\def\o{o}%
\def\SS{SS}%
\def\ss{ss}%
\def\th{th}%
\def\TH{TH}%
\def\dh{dh}%
\def\DH{DH}%
}
\@ifl@t@r\fmtversion{2019/10/01}
{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}
\newcommand{\@gls@removedoption}[1]{%
\PackageError{glossaries}{obsolete package option ‘#1’ has been
removed. Rollback required or use a newer alternative}
{Try \string\usepackage[#1]{glossaries}[=v4.46]}%
}
\define@boolkey[gls]{sanitize}{description}[true]{%
\GlossariesWarning{sanitize={description} package option deprecated}%
\ifgls@sanitize@description
\glssetnoexpandfield{desc}%
\glssetnoexpandfield{descplural}%
\else
\glssetexpandfield{desc}%
\glssetexpandfield{descplural}%
\fi
}
\define@boolkey[gls]{sanitize}{name}[true]{%
\GlossariesWarning{sanitize={name} package option deprecated}%
\ifgls@sanitize@name
\glssetnoexpandfield{name}%
\else
\glssetexpandfield{name}%
\fi
}
}

```

```

\define@boolkey[glS]{sanitize}{symbol}[true]{%
  \GlossariesWarning{sanitize={symbol} package option deprecated}%
  \ifglS@sanitize@symbol
    \glSsetnoexpandfield{symbol}%
    \glSsetnoexpandfield{symbolplural}%
  \else
    \glSsetexpandfield{symbol}%
    \glSsetexpandfield{symbolplural}%
  \fi
}
\define@boolkey{glossaries.sty}[glS]{sanitizesort}[true]{%
  \ifglSsanitizesort
    \glSsetnoexpandfield{sortvalue}%
    \renewcommand*{\@glS@noidx@setsanitizesort}{%
      \glSsanitizesorttrue
      \glSsetnoexpandfield{sortvalue}%
    }%
  \else
    \glSsetexpandfield{sortvalue}%
    \renewcommand*{\@glS@noidx@setsanitizesort}{%
      \glSsanitizesortfalse
      \glSsetexpandfield{sortvalue}%
    }%
  \fi
}
\glSsanitizesorttrue
\glSsetnoexpandfield{sortvalue}%
\newcommand*{\@glS@noidx@setsanitizesort}{%
  \glSsanitizesortfalse
  \glSsetexpandfield{sortvalue}%
}
\define@choicekey[glS]{sanitize}{sort}{true,false}[true]{%
  \setbool{glSsanitizesort}{#1}%
  \ifglSsanitizesort
    \glSsetnoexpandfield{sortvalue}%
  \else
    \glSsetexpandfield{sortvalue}%
  \fi
  \GlossariesWarning{sanitize={sort} package option
    deprecated. Use sanitizesort instead}%
}
\define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
  \ifthenelse{\equal{#1}{none}}{%
    {%
      \GlossariesWarning{sanitize package option deprecated}%
      \glSsetexpandfield{name}%
      \glSsetexpandfield{symbol}%
      \glSsetexpandfield{symbolplural}%
      \glSsetexpandfield{desc}%
      \glSsetexpandfield{descplural}%
    }
  }
}

```

```

}%
{%
  \setkeys[glS]{sanitizE}{#1}%
}%
}
\newif\ifglstranslate
\newcommand*\@glS@usetranslator{%
  \@ifpackageloaded{polyglossia}%
  {%
    \let\glSifusetranslator\@secondoftwo
  }%
  {%
    \@ifpackageloaded{babel}%
    {%
      \IfFileExists{translator.sty}%
      {%
        \RequirePackage{translator}%
        \let\glSifusetranslator\@firstoftwo
      }%
      {}%
    }%
    {}%
  }%
}
\newcommand{\glSifusedtranslatorDict}[3]{%
  \glSifusetranslator
  {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
  {#3}%
}
\@glS@declareoption{notranslate}{%
  \glstranslatefalse
  \let\@glS@usetranslator\relax
  \let\glSifusetranslator\@secondoftwo
}
\define@choicekey{glossaries.sty}{translate}{%
  [\glS@translate@val\glS@translate@nr]%
  {true,false,babel}[true]%
  {%
    \ifcase\glS@translate@nr\relax
    \glstranslatetrue
    \renewcommand*\@glS@usetranslator{%
      \@ifpackageloaded{polyglossia}%
      {%
        \let\glSifusetranslator\@secondoftwo
      }%
      {}%
    }%
    \@ifpackageloaded{babel}%
    {%
      \IfFileExists{translator.sty}%
      {%

```

```

        \RequirePackage{translator}%
        \let\glsifusetranslator\@firstoftwo
    }%
    {}%
} %
{}%
} %
} %
\or
\glstranslatefalse
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\or
\glstranslatetrue
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\fi
}
\glstranslatefalse
\let\glsifusetranslator\@secondoftwo
\@ifpackageloaded{translator}%
{%
\glstranslatetrue
\let\glsifusetranslator\@firstoftwo
}%
{}%
\@for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
{
\@ifpackageloaded{\gls@thissty}%
{%
\glstranslatetrue
\endfortrue
}%
{}%
}
}
\newcommand{\@gls@requested@languages}{}
\define@key{glossaries.sty}{languages}{%
\renewcommand{\@gls@requested@languages}{#1}%
\glstranslatetrue
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
}
\define@boolkey{glossaries.sty}[gls]{indexonlyfirst}[true]{}
\glsindexonlyfirstfalse
\define@boolkey{glossaries.sty}[gls]{hyperfirst}[true]{}
\glshyperfirsttrue
\newcommand*{\@gls@setacrstyle}{}
\define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{}
\ifglsacrfootnote

```



```

    \@gls@removedoption{footnote}%
  \fi
}
\define@boolkey{glossaries.sty}[glsacr]{description}[true]{%
  \ifglsacrdescription
    \@gls@removedoption{description}%
  \fi
}
\define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{%
  \ifglsacrsmallcaps
    \@gls@removedoption{smallcaps}%
  \fi
}
\define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{%
  \ifglsacrsmaller
    \@gls@removedoption{smaller}%
  \fi
}
\define@boolkey{glossaries.sty}[glsacr]{dua}[true]{%
  \ifglsacrdua
    \@gls@removedoption{dua}%
  \fi
}
\define@boolkey{glossaries.sty}[glsacr]{shortcuts}[true]{}
\newcommand*{\glsorder}{word}
\newcommand*{\@glsorder}[1]{}
\define@choicekey{glossaries.sty}{order}{word,letter}{%
  \def\glsorder{#1}}
\newif\ifglsxindy
\glsxindyfalse
\@gls@declareoption{makeindex}{\glsxindyfalse}
\define@boolkey[gls]{xindy}[glsnumbers][true]{}
\gls@xindy@glsnumberstrue
\def\@xdy@main@language{\@languagename}%
\newcommand*{\gls@xdy@set@lang}[1]{%
  \ifcsundef{\@xdy@#1@language}%
    {%
      \ifcsundef{\@xdy@\glsdefaulttype @language}
        {\let\@gls@langmod\@xdy@main@language}%
        {\letcs\@gls@langmod{\@xdy@\glsdefaulttype @language}}}%
    }%
  {\letcs\@gls@langmod{\@xdy@#1@language}}%
}
\define@key[gls]{xindy}{language}{\def\@xdy@main@language{#1}}
\ifdefvoid\inputencodingname
  {\def\gls@codepage{utf8}}%
  {\def\gls@codepage{\inputencodingname}}
\define@key[gls]{xindy}{codepage}{\def\gls@codepage{#1}}
\define@key{glossaries.sty}{xindy}[]{}
  \glsxindytrue

```

```

\setkeys[glS]{xindy}{#1}%
}
\@glS@declareoption{xindygloss}{%
  \glSxindytrue
}
\@glS@declareoption{xindynoglsnumbers}{%
  \glSxindytrue
  \glS@xindy@glSnumbersfalse
}
\providecommand{\@domakeglossaries}[1]{\if@filesw #1\fi}
\@glS@declareoption{disablemakegloss}{%
  \ifdefequal\makeglossaries\@no@makeglossaries
  {%
    \GlossariesWarning{Option ‘disablemakegloss’ has no effect
      (\string\makenoidxglossaries\space has already been used)}%
  }%
  {%
    \ifdefequal\@makeglossary\@gobble
    {%
      \GlossariesWarning{Option ‘disablemakegloss’ has no effect
        (\string\makeglossaries\space has already been used)}%
    }%
  }%
  \renewcommand{\@domakeglossaries}[1]{%
    \PackageInfo{glossaries}{\string\makeglossaries\space and
      \string\makenoidxglossaries\space have been disabled}%
  }%
}
}
\@glS@declareoption{restoremakegloss}{%
  \ifdefequal\makeglossaries\@no@makeglossaries
  {%
    \GlossariesWarning{Option ‘restoremakegloss’ has no effect
      (\string\makenoidxglossaries\space has already been used)}%
  }%
  {%
    \ifdefequal\@makeglossary\@gobble
    {%
      \GlossariesWarning{Option ‘restoremakegloss’ has no effect
        (\string\makeglossaries\space has already been used)}%
    }%
  }%
  \PackageInfo{glossaries}{\string\makeglossaries\space and
    \string\makenoidxglossaries\space have been restored}%
  \let\@domakeglossaries\@firstofone
}
}
\newcommand*{\@do@write@glSlabels}{%

```

```

\AtEndDocument{\do@write@glslabels}%
\let\do@write@glslabels\relax
}
\newcommand*\do@write@glslabels{%
  \newwrite\@gls@labelsfile
  \immediate\openout\@gls@labelsfile=\jobname.glslabels
  \protected@edef\@gls@fulllist{\@glo@types}%
  \ifdefempty{\@ignored@glossaries}{}%
    {\protected@eappto\@gls@fulllist{\@ignored@glossaries}}%
  \forallglsentries[\@gls@fulllist]{\@glsentry}%
  {%
    \ifdefempty{\@glsentry}{}%
    {%
      \ifx\do@write@glslabelsname@gobble
        \def\@gls@thisentryname{%
        \else
          \letcs{\@gls@thisentryname}{glo@\@glsdetoklabel{\@glsentry}@name}%
          \@onelevel@sanitize\@gls@thisentryname
        \fi
        \immediate\write\@gls@labelsfile
          {\@glsentry\do@write@glslabelsname{\@gls@thisentryname}}%
      }%
    }%
  \immediate\closeout\@gls@labelsfile
}
\newcommand*\do@write@glslabelsname}[1]{%
\begingroup
\catcode'\^^I12
\gdef\@write@glslabelsname#1{\^^I#1}
\endgroup
\@gls@declareoption{writeglslabels}{\do@write@glslabels
\let\do@write@glslabelsname@gobble}
\@gls@declareoption{writeglslabelnames}{\do@write@glslabels
\let\do@write@glslabelsname@write@glslabelsname}
\newif\ifglsautomake
\newcommand{\gls@automake@nr}{1}
\newcommand{\gls@automake@makegloss}{%
\newcommand{\gls@automake@option}{automake=\gls@automake@val}
\newcommand*\@gls@do@automake@err}{%
  \PackageError{glossaries}{You must use
  \string\makeglossaries\space with \gls@automake@option}
  {%
    Either remove the \gls@automake@option\space setting or
    add \string\makeglossaries\space to your document preamble.%
  }%
}%
\define@choicekey{glossaries.sty}{automake}%
  [\gls@automake@val\gls@automake@nr]{true,false,immediate,makegloss,lite,delayed}[immediate]{%
  \ifcase\gls@automake@nr\relax
    \GlossariesWarning{automake=true deprecated. Use automake=delayed for a delayed write}%

```

```

\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomakefalse
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{makeglossaries}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
\or
\def\gls@automake@nr{0}%
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\fi
\renewcommand{\gls@automake@option}{automake=\gls@automake@val}%
\ifglsautomake
\let\@gls@doautomake\@gls@do@automake@err
\else
\renewcommand*\@gls@doautomake{}%
\fi
}
\glsautomakefalse
\newcommand*\@gls@doautomake{}
\AtEndDocument{\@gls@doautomake}
\@gls@declareoption{automakegloss}{%
\glsautomaketrue
\renewcommand{\gls@automake@option}{automakegloss}%
\def\gls@automake@val{makegloss}%
\def\gls@automake@nr{3}%
\renewcommand{\gls@automake@makegloss}{makeglossaries}%
\let\@gls@doautomake\@gls@do@automake@err
}
\@gls@declareoption{automakeglosslite}{%
\glsautomaketrue
\renewcommand{\gls@automake@option}{automakeglosslite}%
\def\gls@automake@val{lite}%
\def\gls@automake@nr{4}%
\renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
\let\@gls@doautomake\@gls@do@automake@err
}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
\ifgls savewrites
\renewcommand*\@gls@writefiles{\@gls@writefiles}%
\else
\let\@gls@writefiles\@empty

```

```

\fi
}
\glssavewritesfalse
\let\glswritefiles\@empty
\define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{%
  \ifbool{glscompatible-3.07}%
  {%
    \@gls@removedoption{compatible-3.07}%
  }%
  {}%
}
\boolfalse{glscompatible-3.07}
\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
  \ifbool{glscompatible-2.07}%
  {%
    \@gls@removedoption{compatible-2.07}%
  }%
  {}%
}
\boolfalse{glscompatible-2.07}
\let\gls@original@makeglossary\makeglossary
\let\gls@original@glossary\glossary
\newcommand*{\@gls@override@glossary}[1][main]{%
  \GlossariesWarning{Use of \string\glossary\space with
glossaries.sty is deprecated. \MessageBreak Indexing should be performed
with the user level \MessageBreak commands, such as \string\gls\space or
\string\glsadd. If you need the \MessageBreak original definition of
\string\glossary\space use the package \MessageBreak options
kernelglossredefs=false (to restore the \MessageBreak former definition of
\string\glossary) and nomain (if the \MessageBreak file extensions cause a
conflict)}}%
\gls@glossary{#1}%
}
\define@choicekey{glossaries.sty}{kernelglossredefs}{%
  [\gls@kernelglossredefs@val\gls@kernelglossredefs@nr]{true,false,nowarn}[true]{%
  {%
    \ifcase\gls@kernelglossredefs@nr\relax
    \def\glossary{\@gls@override@glossary}%
    \def\makeglossary{%
      \GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is deprecated. Use \string\makeglossaries\space
instead. If you need the original definition of
\string\makeglossary\space use the package options
kernelglossredefs=false (to prevent redefinition of
\string\makeglossary) and nomain (if the file extensions cause a
conflict)}}%
      \makeglossaries
    }%
    \or
    \let\glossary\gls@original@glossary

```

```

\let\makeglossary\gls@original@makeglossary
\or
\def\makeglossary{\makeglossaries}%
\renewcommand*\@gls@override@glossary}[1][main]{%
  \gls@glossary{##1}%
}%
\fi
}
\@gls@declareoption{symbols}{%
  \let\@gls@do@symbolsdef\@gls@symbolsdef
}
\newcommand*\@gls@do@symbolsdef{}
\newcommand*\@gls@symbolsdef{%
  \newglossary[slg]{symbols}{sls}{slo}{\glsymbolsgroupname}%
  \newcommand*\@printsymbols}[1][\printglossary[type=symbols,##1]]{%
  \newcommand*\@gls@tr@set@symbols@toctitle}{%
    \translatelet{\glossarytoctitle}{Symbols (glossaries)}%
  }%
}%
\@gls@declareoption{numbers}{%
  \let\@gls@do@numbersdef\@gls@numbersdef
}
\newcommand*\@gls@do@numbersdef{}
\newcommand*\@gls@numbersdef{%
  \newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
  \newcommand*\@printnumbers}[1][\printglossary[type=numbers,##1]]{%
  \newcommand*\@gls@tr@set@numbers@toctitle}{%
    \translatelet{\glossarytoctitle}{Numbers (glossaries)}%
  }%
}%
\@gls@declareoption{index}{%
  \ifx\@gls@do@indexdef\@empty
    \let\@gls@do@indexdef\@gls@indexdef
  \fi
}
\@gls@declareoption{noglossaryindex}{%
  \let\@gls@do@indexdef\relax
}
\newcommand*\@gls@do@indexdef{}
\newcommand*\@gls@indexdef{%
  \newglossary[ilg]{index}{ind}{idx}{\indexname}%
  \newcommand*\@printindex}[1][\printglossary[type=index,##1]]{%
  \newcommand*\@newterm}[2][%
    \newglossaryentry{##2}%
    {type={index},name={##2},description={\nopostdesc},##1}}
  \let\@gls@do@indexdef\relax
}%
\define@choicekey{glossaries.sty}{mfirstuc}{%
  [\gls@mfirstuc@val\gls@mfirstuc@nr]%
  {expanded,unexpanded}%
}

```

```

{%
  \ifcase\gls@mfirstuc@nr
  % expanded
  \renewcommand*\glsmakefirstuc}[1]{\MFUsentencecase{#1}}%
  \or
  % unexpanded
  \renewcommand*\glsmakefirstuc}[1]{\MFUsentencecase{\unexpanded{#1}}}%
  \fi
}
\@for\CurrentOption :=\@declaredoptions\do{%
  \ifx\CurrentOption\@empty
  \else
  \@expandtwoargs
  \in@ {,\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
  \ifin@
  \@use@ption
  \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
  \fi
  \fi
}
\ProcessOptionsX
\ifglsautomake
\ifdef\directlua{\RequirePackage{shellesc}}{}
\fi
\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,notranslate,languages,nolong,nosuper,notree,nostyles,%
nomain,noglossaryindex}
\newcommand*\setupglossaries}[1]{%
  \renewcommand*\@gls@setacrstyle}{}%
  \ifglsacrshortcuts
  \def\@gls@setupshortcuts{\glsacrshortcutstrue}%
  \else
  \def\@gls@setupshortcuts{%
    \ifglsacrshortcuts
    \DefineAcronymSynonyms
  \fi
  }%
  \fi
  \glsacrshortcutsfalse
  \let\@gls@do@numbersdef\relax
  \let\@gls@do@symbolssdef\relax
  \let\@gls@do@indexdef\relax
  \let\@gls@do@acronymsdef\relax
  \ifglsubentrycounter
  \let\@gls@doentrycounterdef\relax
  \else
  \let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
  \fi
  \ifglsubentrycounter

```

```

\let\@gls@dosomeentrycounterdef\relax
\else
\let\@gls@dosomeentrycounterdef\@gls@define@glossarysubentrycounter
\fi
\setkeys{glossaries.sty}{#1}%
\@gls@setacrstyle
\@gls@setupshortcuts
\@gls@do@acronymsdef
\@gls@do@numbersdef
\@gls@do@symbolssdef
\@gls@do@indexdef
\@gls@doentrycounterdef
\@gls@dosomeentrycounterdef
}
\ifthenelse{\equal{\glscounter}{section}}%
{%
\ifcsundef{chapter}{}%
{%
\let\@gls@old@chapter\@chapter
\def\@chapter[#1]#2{\@gls@old@chapter[#1]}{#2}%
\ifcsundef{hyperdef}{\hyperdef{section}{\thesection}}}%
}%
}%
{}
\newcommand*\@gls@onlypremakeg{}
\newcommand*\@onlypremakeg[1]{%
\ifx\@gls@onlypremakeg\@empty
\def\@gls@onlypremakeg{#1}%
\else
\expandafter\toks\expandafter{\@gls@onlypremakeg}%
\edef\@gls@onlypremakeg{\the\toks@,\noexpand#1}%
\fi
}
\newcommand*\@disable@onlypremakeg{%
\@for\@thiscs:=\@gls@onlypremakeg\do{%
\expandafter\@disable@premakecs\@thiscs%
}}
\newcommand*\@disable@premakecs[1]{%
\def#1{\PackageError{glossaries}{\string#1\space may only be
used before \string\makeglossaries}{You can't use
\string#1\space after \string\makeglossaries}}%
}
\providecommand*\glossaryname{Glossary}
\providecommand*\acronymname{Acronyms}
\newcommand*\glssettoctitle[1]{%
\def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}
\providecommand*\entryname{Notation}
\providecommand*\descriptionname{Description}
\providecommand*\symbolname{Symbol}
\providecommand*\pagelistname{Page List}

```



```

\providecommand*\glssymbolsgroupname}{Symbols}
\providecommand*\glsnumbersgroupname}{Numbers}
\newcommand*\glspluralsuffix}{s}
\newcommand*\glsacrpluralsuffix}{\glspluralsuffix}
\newcommand*\glsupacrpluralsuffix}{\glstextup{\glsacrpluralsuffix}}
\providecommand*\seename}{see}
\providecommand*\andname}{\&}
\newcommand*\RequireGlossariesLang}[1]{%
  \ifundefined{ver@glossaries-#1.ldf}{\input{glossaries-#1.ldf}}{}%
}
\newcommand*\ProvidesGlossariesLang}[1]{%
  \ProvidesFile{glossaries-#1.ldf}%
}
\newcommand*\addglossarytocaptions}[1]{}
\ifglstranslate
  \RequirePackage{tracklang}[2019/11/30]
  \ifvoid\@gls@requested@languages{}%
  {%
    \@for\@gls@thislang:=\@gls@requested@languages\do{%
      \TrackIfKnownLanguage{\@gls@thislang}{}%
      {\GlossariesWarning{unknown language ‘\@gls@thislang’}}%
    }%
  }%
  \@gls@usetranslator
  \@ifpackageloaded{translator}
  {%
    \ifboolexpr
    {
      test {\ifdefstring{\trans@languages}{English}}
      and not
      test {\ifdefstring{bbl@loaded}{english}}
    }
    {%
      \let\glsifusetranslator\@secondoftwo
    }%
    {%
      \usedictionary{glossaries-dictionary}%
      \renewcommand*\addglossarytocaptions}[1]{%
        \ifcsundef{captions#1}{}%
        {%
          \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
          \expandafter\toks@\expandafter{\@gls@tmp
            \renewcommand*\glossaryname}{\translate{Glossary}}%
          }%
          \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
        }%
      }%
    }%
  }%
}{}%

```

```

\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
  }%
}%
{}%
\glsifusetranslator
{%
  \renewcommand*\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*\glossaryname{\translate{Glossary}}%
  \renewcommand*\acronymname{\translate{Acronyms}}%
  \renewcommand*\entryname{\translate{Notation (glossaries)}}%
  \renewcommand*\descriptionname{%
    \translate{Description (glossaries)}}%
  \renewcommand*\symbolname{\translate{Symbol (glossaries)}}%
  \renewcommand*\pagelistname{%
    \translate{Page List (glossaries)}}%
  \renewcommand*\glssymbolsgroupname{%
    \translate{Symbols (glossaries)}}%
  \renewcommand*\glsnumbersgroupname{%
    \translate{Numbers (glossaries)}}%
}%
}{}%
\fi
\DeclareRobustCommand*\nopostdesc{}
\newcommand*\@nopostdesc{%
  \let\org@glspostdescription\glspostdescription
  \def\glspostdescription{%
    \let\glspostdescription\org@glspostdescription}%
}
\newcommand*\@no@post@desc{\nopostdesc}
\newcommand*\glspar{\par}
\newcommand*\setStyleFile[1]{%
  \renewcommand*\gls@istfilebase}{#1}%

```

```

\ifglxindy
  \def\istfilename{\gls@istfilebase.xdy}
\else
  \def\istfilename{\gls@istfilebase.ist}
\fi
}
\@onlypremakeg\setStyleFile
\ifglxindy
  \def\istfilename{\gls@istfilebase.xdy}
\else
  \def\istfilename{\gls@istfilebase.ist}
\fi
\newcommand*\gls@istfilebase{\jobname}
\newcommand*\@istfilename[1]{
\newcommand*\glscompositor{.}
\newcommand*\glsSetCompositor[1]{%
  \renewcommand*\glscompositor{#1}}
\@onlypremakeg\glsSetCompositor
\newcommand*\@glsAlphacompositor{\glscompositor}
\ifglxindy
  \newcommand*\glsSetAlphaCompositor[1]{%
    \renewcommand*\@glsAlphacompositor{#1}}
\else
  \newcommand*\glsSetAlphaCompositor[1]{%
    \glsnoxywarning\glsSetAlphaCompositor}
\fi
\@onlypremakeg\glsSetAlphaCompositor
\newcommand*\gls@suffixF{}
\newcommand*\glsSetSuffixF[1]{%
  \renewcommand*\gls@suffixF{#1}}
\@onlypremakeg\glsSetSuffixF
\newcommand*\gls@suffixFF{}
\newcommand*\glsSetSuffixFF[1]{%
  \renewcommand*\gls@suffixFF{#1}}
}
\ifcsundef{hyperlink}%
{%
  \newcommand*\glsnumberformat[1]{#1}%
}%
{%
  \newcommand*\glsnumberformat[1]{\glshypernumber{#1}}%
}
\newcommand{\glslocationstoencap}[3]{\csuse{#2}{\csuse{#1}{#3}}}
\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*\glossarypreamble{%
  \csuse{@glossarypreamble@currentglossary}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaultttype]{%
  \ifglossaryexists*#1}%

```

```

{\csgdef{@glossary preamble@#1}{#2}}%
{\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*{\glossary postamble}{}
\newcommand*{\glossary section}[2][\@gls@title]{%
\def\@gls@title{#2}%
\ifcsundef{phantomsection}%
{%
\@glossary section{#1}{#2}%
}%
{%
\@pglossary section{#1}{#2}%
}%
\glsglossary mark{\glossary toctitle}%
}
\ifcsundef{glossary mark}%
{%
\newcommand{\glsglossary mark}[1]{\glossary mark{#1}}
}%
{%
\@ifclassloaded{memoir}
{%
\newcommand{\glsglossary mark}[1]{%
\ifglsucmark
\markboth{\memUchead{#1}}{\memUchead{#1}}%
\else
\markboth{#1}{#1}%
\fi
}
}%
{%
\newcommand{\glsglossary mark}[1]{%
\ifglsucmark
\@mkboth{\glsupercase{#1}}{\glsupercase{#1}}%
\else
\@mkboth{#1}{#1}%
\fi
}
}
}
\providecommand{\glossary mark}[1]{%
\ifglsucmark
\@mkboth{\glsupercase{#1}}{\glsupercase{#1}}%
\else
\@mkboth{#1}{#1}%
\fi
}
\newcommand*{\setglossary section}[1]{%
\setkeys{glossaries.sty}{section=#1}}
\newcommand*{\@glossary section}[2]{%

```

```

\ifdefempty\@@glossarysecstar
{%
  \csname\@@glossarysec\endcsname[#1]{#2}%
}%
{%
  \csname\@@glossarysec\endcsname*{#2}%
  \@gls@toc{#1}{\@@glossarysec}%
}%
\@@glossaryseclabel
}
\newcommand*\@p@glossarysection}[2]{%
  \gls@clearpage
  \phantomsection
  \ifdefempty\@@glossarysecstar
  {%
    \csname\@@glossarysec\endcsname{#2}%
  }%
  {%
    \@gls@toc{#1}{\@@glossarysec}%
    \csname\@@glossarysec\endcsname*{#2}%
  }%
  \@@glossaryseclabel
}
\newcommand*\gls@doclearpage{%
  \ifthenelse{\equal{\@@glossarysec}{chapter}}{%
    {%
      \ifcsundef{cleardoublepage}%
      {%
        \clearpage
      }%
      {%
        \ifcsdef{if@openright}%
        {%
          \if@openright
            \cleardoublepage
          \else
            \clearpage
          \fi
        }%
      }%
      \cleardoublepage
    }%
  }%
  {}%
}
\newcommand*\gls@clearpage{\gls@doclearpage}
\newcommand*\@gls@toc}[2]{%
  \ifglstoc
  \ifglsnumberline

```

```

        \addcontentsline{toc}{#2}{\protect\numberline{#1}}%
    \else
        \addcontentsline{toc}{#2}{#1}%
    \fi
\fi
}
\newcommand*\glsnoindeywarning}[1]{%
    \GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
}
\newcommand*\glsnomakeindexwarning}[1]{%
    \GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%
}
\ifglsxindy
    \edef\xdyattributes{\string"default\string"}%
\fi
\ifglsxindy
    \edef\xdyattributelist{}%
\fi
\ifglsxindy
    \def\xdylocref{}
\fi
\newcommand*\@gls@ifinlist}[4]{%
    \def\@do@ifinlist##1,#1,##2\end@do@ifinlist{%
        \def\@gls@listsuffix{##2}%
        \ifx\@gls@listsuffix\@empty
            #4%
        \else
            #3%
        \fi
    }%
    \@do@ifinlist,#2,#1,\end@do@ifinlist
}
\ifglsxindy
    \newcommand*\@xdycounters{\@glscounter}
    \newcommand*\GlsAddXdyCounters[1]{%
        \@for\@gls@ctr:=#1\do{%
            \edef\@do@addcounter{%
                \noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
                {%
                    \noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
                        \noexpand\@gls@ctr}%
                }%
            }%
            \@do@addcounter
        }
    }
    \@onlypremakeg\GlsAddXdyCounters
\else
    \newcommand*\GlsAddXdyCounters[1]{%
        \glsnoindeywarning\GlsAddXdyAttribute

```

```

}
\fi
\newcommand*\@disabled@glSaddxdycounters{%
  \PackageError{glossaries}{\string\GlsAddXdyCounters\space
    can't be used after \string\GlsAddXdyAttribute}{Move all
    occurrences of \string\GlsAddXdyCounters\space before the first
    instance of \string\GlsAddXdyAttribute}%
}
\ifglSxindy
  \newcommand*\@glSaddxdyattribute[2]{%
    \edef\@xdyattributes{\@xdyattributes ^^J
      \string"#2#1\string"}%
    \expandafter\toks@\expandafter{\@xdylocref}%
    \edef\@xdylocref{\the\toks@ ^^J%
      (markup-locref
      :open \string"\glStildechar n%
        \expandafter\string\csname glSX#2X#1\endcsname
        \string" ^^J
      :close \string"\string" ^^J
      :attr \string"#2#1\string")}%
    \expandafter\gdef\csname glSX#2X#1\endcsname##1##2{%
      \setentrycounter[##1][##2]\csname #1\endcsname{##2}%
    }%
  }
\newcommand*\GlsAddXdyAttribute[1]{%
  \ifx\@xdyattributelist\@empty
    \edef\@xdyattributelist{#1}%
  \else
    \edef\@xdyattributelist{\@xdyattributelist,#1}%
  \fi
  \edef\@xdyattributes{\@xdyattributes ^^J
    \string"#1\string"}%
  \@for\@this@counter:=\@xdycounters\do{%
    \protected@edef\glS@do@addxdyattribute{%
      \noexpand\@glSaddxdyattribute{#1}{\@this@counter}%
    }
    \glS@do@addxdyattribute
  }%
  \let\GlsAddXdyCounters\@disabled@glSaddxdycounters
}
\@onlypremakeg\GlsAddXdyAttribute
\else
  \newcommand*\GlsAddXdyAttribute[1]{%
    \glSnoxindywarning\GlsAddXdyAttribute}
\fi
\ifglSxindy
\newcommand*{\@glS@addpredefinedattributes}{%
  \GlsAddXdyAttribute{glSnumberformat}
  \GlsAddXdyAttribute{textrm}
  \GlsAddXdyAttribute{textsf}

```

```

\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glshypernumber}
\GlsAddXdyAttribute{hyperrm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hypertt}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}
\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}
\GlsAddXdyAttribute{glsignore}
}
\else
\let\@gls@addpredefinedattributes\relax
\fi
\def\@xdyuseralphabets{}
\ifglxindy
\newcommand*{\GlsAddXdyAlphabet}[2]{%
\edef\@xdyuseralphabets{%
\@xdyuseralphabets ~^J
(define-alphabet "#1" (#2))}}
\else
\newcommand*{\GlsAddXdyAlphabet}[2]{%
\glsnoxywarning\GlsAddXdyAlphabet}
\fi
\ifglxindy
\newcommand*{\@gls@xdy@locationlist}{%
roman-page-numbers,%
Roman-page-numbers,%
arabic-page-numbers,%
alpha-page-numbers,%
Alpha-page-numbers,%
Appendix-page-numbers,%
arabic-section-numbers%
}
\protected@edef\@gls@roman{\@roman{0}\string"
\string"roman-numbers-lowercase\string" :sep \string"}%
\@onelevel@sanitize\@gls@roman
\edef\@gls@tmp{\string" \string"roman-numbers-lowercase\string"
:sep \string"}%
\@onelevel@sanitize\@gls@tmp

```



```

\ifx\@gls@tmp\@gls@roman
  \expandafter
    \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
      \string"roman-numbers-lowercase\string"%
    }%
\else
  \expandafter
    \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{
      :sep \string"\@gls@roman\string"%
    }%
\fi
\expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
  \string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
  \string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
  \string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
  \string"ALPHA\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
  \string"ALPHA\string"
  :sep \string"\@gls@Alphacompositor\string"
  \string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
  \string"arabic-numbers\string"
  :sep \string"\glscompositor\string"
  \string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglxindy
  \newcommand\@xdycrossrefhook{}
\fi
\ifglxindy
  \newcommand*\GlsAddXdyLocation}[3][[]]{%
    \def\@gls@tmp[#1]{%
      \ifx\@gls@tmp\@empty
        \edef\@xdyuserlocationdefs{%
          \@xdyuserlocationdefs ^^J%
          (define-location-class \string"#2\string"^^J\space\space
            \space(:sep \string"{}\glsopenbrace\string" #3
              :sep \string"\glsclosebrace\string"))
        }%
      \else

```

```

\edef\@xdyuserlocationdefs{%
  \@xdyuserlocationdefs ^^J%
  (define-location-class \string"#2\string"^^J\space\space
  \space(:sep "\glsoopenbrace"
    #1
    :sep "\glsclosebrace\glsoopenbrace" #3
    :sep "\glsclosebrace"))
}%
\fi
\edef\@xdyuserlocationnames{%
  \@xdyuserlocationnames^^J\space\space\space
  \string"#2\string"%
}
\@onlypremakeg\GlsAddXdyLocation
\else
  \newcommand*\GlsAddXdyLocation[2]{%
    \glsnoinindywarning\GlsAddXdyLocation}
\fi
\ifglxsindy
  \def\@xdylocationclassorder{^^J\space\space\space
    \string"roman-page-numbers\string"^^J\space\space\space
    \string"arabic-page-numbers\string"^^J\space\space\space
    \string"arabic-section-numbers\string"^^J\space\space\space
    \string"alpha-page-numbers\string"^^J\space\space\space
    \string"Roman-page-numbers\string"^^J\space\space\space
    \string"Alpha-page-numbers\string"^^J\space\space\space
    \string"Appendix-page-numbers\string"
    \@xdyuserlocationnames^^J\space\space\space
    \string"see\string"
  }
\fi
\ifglxsindy
  \newcommand*\GlsSetXdyLocationClassOrder[1]{%
    \def\@xdylocationclassorder{#1}}
\else
  \newcommand*\GlsSetXdyLocationClassOrder[1]{%
    \glsnoinindywarning\GlsSetXdyLocationClassOrder}
\fi
\ifglxsindy
  \def\@xdysortrules{}
\fi
\ifglxsindy
  \newcommand*\GlsAddSortRule[2]{%
    \expandafter\toks@\expandafter{\@xdysortrules}%
    \protected@edef\@xdysortrules{\the\toks@ ^^J
      (sort-rule \string"#1\string" \string"#2\string"))%
  }
\else
  \newcommand*\GlsAddSortRule[2]{%
    \glsnoinindywarning\GlsAddSortRule}

```

```

\fi
\ifglxindy
  \def\@xdyrequiredstyles{tex}
\fi
\ifglxindy
  \newcommand*\GlsAddXdyStyle[1]{%
    \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
  \newcommand*\GlsAddXdyStyle[1]{%
    \glsnoxywarning\GlsAddXdyStyle}
\fi
\ifglxindy
  \newcommand*\GlsSetXdyStyles[1]{%
    \edef\@xdyrequiredstyles{#1}}
\else
  \newcommand*\GlsSetXdyStyles[1]{%
    \glsnoxywarning\GlsSetXdyStyles}
\fi
\def\@xdy@language#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
    \ifglossaryexists{#1}{%
      \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
    }{%
      \PackageError{glossaries}{Can't set language type for
        glossary type '#1' --- no such glossary}{%
        You have specified a glossary type that doesn't exist}}
\else
  \newcommand*\GlsSetXdyLanguage[2][]{%
    \glsnoxywarning\GlsSetXdyLanguage}
\fi
\def\@gls@codepage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyCodePage[1]{%
    \renewcommand*\@gls@codepage{#1}%
  }
  \AtBeginDocument{%
    \ifx\@gls@codepage\@empty
      \ifpackage@loaded{fontspec}{\def\@gls@codepage{utf8}}}%
  \fi
}
\else
  \newcommand*\GlsSetXdyCodePage[1]{%
    \glsnoxywarning\GlsSetXdyCodePage}
\fi
\ifglxindy
  \ifglx@xindy@glsnumbers
    \def\@xdy@lettergroups{(define-letter-group
      \string"glxnumbers\string"^^J\space\space\space
      :prefixes (\string"0\string" \string"1\string"

```

```

        \string"2\string" \string"3\string" \string"4\string"
        \string"5\string" \string"6\string" \string"7\string"
        \string"8\string" \string"9\string")^^J\space\space\space
        \@xdynumbergrouporder)}
\else
  \def\@xdylettergroups{}
\fi
\fi
\newcommand*\GlsAddLetterGroup[2]{%
  \expandafter\toks@\expandafter{\@xdylettergroups}%
  \protected@edef\@xdylettergroups{\the\toks@^^J%
    (define-letter-group \string"#1\string"^^J\space\space\space#2)}%
  }%
\newcommand*\forallglossaries}[3][\@glo@types]{%
  \@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%
}

\newcommand*\forallacronyms}[2]{%
  \@for#1:=\@glsacronymlists\do{\ifx#1\@empty\else#2\fi}%
}

\newcommand*\forglentries}[3][\glsdefaulttype]{%
  \protected@edef\@glo@list{\csname glolist@#1\endcsname}%
  \@for#2:=\@glo@list\do
  {%
    \ifdefempty{#2}{#3}%
  }%
}

\newcommand*\forallglentries}[3][\@glo@types]{%
  \expandafter\forallglossaries\expandafter[#1]{\@thisglo@}%
  {%
    \forglentries[\@thisglo@]{#2}{#3}%
  }%
}

\newcommand{\ifglossaryexists}{%
  \@ifstar\s@ifglossaryexists\@ifglossaryexists
}

\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{@glo@type@#1@out}{#3}{#2}%
}

\newcommand{\s@ifglossaryexists}[3]{%
  \ifcsundef{glolist@#1}{#3}{#2}%
}

\newcommand*\glsdetoklabel}[1]{#1}
\newcommand*\ifglentryexists}[3]{%
  \ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}

\newcommand*\ifglsused}[3]{%
  \ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}

\newcommand*\glsdoifexists}[2]{%

```

```

\ifglentryexists{#1}{#2}{%
  \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
    has not been defined}{You need to define a glossary entry before you
    can use it.}}%
}
\newcommand{\glsdoifnoexists}[2]{%
  \ifglentryexists{#1}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
      been defined}{}}{#2}%
}
\newcommand{\glsdoifexistsorwarn}[2]{%
  \ifglentryexists{#1}{#2}{%
    \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}%
  }%
}
\newcommand{\glsdoifexistsordo}[3]{%
  \ifglentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
  #3%
}%
}
\newcommand{\doifglossarynoexistsordo}[3]{%
  \ifglossaryexists*{#1}%
  {%
    \PackageError{glossaries}{Glossary type ‘#1’ already exists}{}%
    #3%
  }%
  {#2}%
}
\newrobustcmd{\ifglshaschildren}[3]{%
  \glsdoifexists{#1}%
  {%
    \def\do@glshaschildren{#3}%
    \protected@edef\@gls@thislabel{\glsdetoklabel{#1}}%
    \expandafter\for@glentries\expandafter
      [\csname glo@\@gls@thislabel @type\endcsname]
    {\glo@label}%
    {%
      \letcs@glo@parent{glo@\glo@label @parent}%
      \ifdefequal\@gls@thislabel@glo@parent
        {%
          \def\do@glshaschildren{#2}%
          \@endfortrue
        }%
      }%
    }%
  }%
  \do@glshaschildren
}

```

```

    }%
  }
  \newcommand{\ifglshasparent}[3]{%
    \glsdoifexists{#1}%
    {%
      \ifcsemtyp{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
    }%
  }
  \newcommand*\ifglshasdesc}[3]{%
    \ifcsemtyp{glo@\glsdetoklabel{#1}@desc}%
    {#3}%
    {#2}%
  }
  \newcommand*\ifglshasdescsuppressed}[3]{%
    \ifcsequal{glo@\glsdetoklabel{#1}@desc}{@no@post@desc}%
    {#2}%
    {#3}%
  }
  \newrobustcmd*\ifglshassymbol}[3]{%
    \letcs{\@glo@symbol}{glo@\glsdetoklabel{#1}@symbol}%
    \ifdefempty\@glo@symbol
    {#3}%
    {%
      \ifdefequal\@glo@symbol\@gls@default@value
      {#3}%
      {#2}%
    }%
  }
  \newrobustcmd*\ifglshaslong}[3]{%
    \letcs{\@glo@long}{glo@\glsdetoklabel{#1}@long}%
    \ifdefempty\@glo@long
    {#3}%
    {%
      \ifdefequal\@glo@long\@gls@default@value
      {#3}%
      {#2}%
    }%
  }
  \newrobustcmd*\ifglshasshort}[3]{%
    \letcs{\@glo@short}{glo@\glsdetoklabel{#1}@short}%
    \ifdefempty\@glo@short
    {#3}%
    {%
      \ifdefequal\@glo@short\@gls@default@value
      {#3}%
      {#2}%
    }%
  }
  \newcommand{\ifglsfieldvoid}[4]{\ifcsvoid{glo@\glsdetoklabel{#2}@#1}{#3}{#4}}
  \newrobustcmd*\ifglshasfield}[4]{%

```

```

\glsdoifexists{#2}%
{%
  \letcs{\@glo@thisvalue}{glo@glsetoklabel{#2}@#1}%
  \ifdef\@glo@thisvalue
  {%
    \ifdefempty\@glo@thisvalue
    {%
      #4%
    }%
  }%
  \ifdequal\@glo@thisvalue\@gls@default@value
  {%
    #4%
  }%
  {%
    \let\glscurrentfieldvalue\@glo@thisvalue
    #3%
  }%
}%
}%
{%
  \@gls@fetchfield{\@gls@thisfield}{#1}%
  \ifdef\@gls@thisfield
  {%
    \letcs{\@glo@thisvalue}{glo@glsetoklabel{#2}@ \@gls@thisfield}%
    \ifdefempty\@glo@thisvalue
    {%
      #4%
    }%
    {%
      \ifdequal\@glo@thisvalue\@gls@default@value
      {%
        #4%
      }%
    }%
    \let\glscurrentfieldvalue\@glo@thisvalue
    #3%
  }%
}%
}%
{%
  \GlossariesWarning{Unknown entry field '#1'}%
  #4%
}%
}%
}
\newcommand*{\glscurrentfieldvalue}{ }
\newcommand*{\@glo@types}{,}
\newcommand*{\@gls@provide@newglossary}{%

```

```

\protected@write\@auxout{}\string\providecommand\string\@newglossary[4]{}%
\let\@gls@provide@newglossary\relax
}
\newcommand*{\@gls@defaulttype}{\glsdefaulttype}
\newcommand*{\defglsentryfmt}[2][\glsdefaulttype]{%
\protected@edef\@glo@tmp{#1}%
\ifx\@glo@tmp\@gls@defaulttype
\GlossariesWarning{\string\defglsentryfmt\space attempting to set
a display style for \string\glsdefaulttype\space but there
doesn't seem to be a default glossary. If you have used
'nomain', you need to define an alternative glossary}%
\else
\csgdef{gls@\@glo@tmp @entryfmt}{#2}%
\fi
}
\newcommand*{\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*{\@gls@forbidtext}[1]{%
\ifboolexpr{test {\ifdefstring{#1}{tex}}
or test {\ifdefstring{#1}{TEX}}}
{%
\def#1{nottex}%
\PackageError{glossaries}%
{Forbidden '.tex' extension replaced with '.nottex'}%
{I'm sorry, I can't allow you to do something so reckless.\MessageBreak
Don't use '.tex' as an extension for a temporary file.}%
}%
}%
}%
}
\newcommand*{\gls@gobbleopt}{\new@ifnextchar[{\@gls@gobbleopt}{}]}
\def\@gls@gobbleopt[#1]{
\newcommand*{\newglossary}\@ifstar\s@newglossary\ns@newglossary}
\newcommand*{\s@newglossary}[2]{%
\ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
}
\newcommand*{\ns@newglossary}[5][glg]{%
\doifglossarynoexistsordo{#2}%
{%
\ifundef\glsdefaulttype
{%
\gdef\glsdefaulttype{#2}%
}{}%
\toks@{#2}\protected@edef\@glo@types{\@glo@types\the\toks@,}%
\expandafter\gdef\csname glolist@#2\endcsname{,}%
\expandafter\edef\csname @glotype@#2@log\endcsname{#1}%
\expandafter\edef\csname @glotype@#2@in\endcsname{#3}%
\expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
\expandafter\@gls@forbidtext\csname @glotype@#2@log\endcsname
\expandafter\@gls@forbidtext\csname @glotype@#2@in\endcsname
\expandafter\@gls@forbidtext\csname @glotype@#2@out\endcsname

```



```

\expandafter\def\csname @glotype@#2@title\endcsname{#5}%
\@gls@provide@newglossary
\protected@write\@auxout{}\string\@newglossary{#2}{#1}{#3}{#4}}%
\ifcsundef{gls@#2@entryfmt}%
{%
  \defglsentryfmt[#2]{\glsentryfmt}%
}%
{}%
\@gls@defsortcount{#2}%
\@ifnextchar[{\@gls@setcounter{#2}}%
  {\@gls@setcounter{#2}[\glscounter]}%
}%
{%
  \gls@gobbleopt
}%
}
\newcommand*\altnewglossary}[3]{%
  \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
}
\@onlypreamble{\newglossary}
\@onlypremakeg\newglossary
\newcommand*\@newglossary}[4]{%
\def\@gls@setcounter#1[#2]{%
  \expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
  \ifglsxindy
    \GlsAddXdyCounters{#2}%
  \fi
}
\newcommand*\@gls@getcounter}[1]{%
  \csname @glotype@#1@counter\endcsname
}
\glsdefmain
\@gls@do@acronymsdef
\@gls@do@symbolsdef
\@gls@do@numbersdef
\@gls@do@indexdef
\newcommand*\newignoredglossary}[1]{%
  \ifdefempty\@ignored@glossaries
  {%
    \protected@edef\@ignored@glossaries{#1}%
  }%
  {%
    \protected@eappto\@ignored@glossaries{,#1}%
  }%
  \csgdef{glolist@#1}{,}%
  \ifcsundef{gls@#1@entryfmt}%
  {%
    \defglsentryfmt[#1]{\glsentryfmt}%
  }%
  {}%
}

```

```

\ifdefempty\@gls@nohyperlist
{%
  \renewcommand*\@gls@nohyperlist{#1}%
}%
{%
  \protected@eappto\@gls@nohyperlist{, #1}%
}%
}
\newcommand*\@ignored@glossaries{}
\newcommand*\@ifignoredglossary}[3]{%
  \protected@edef\@gls@igtype{#1}%
  \expandafter\DTLifinlist\expandafter
  {\@gls@igtype}\@ignored@glossaries}{#2}{#3}%
}
\define@key{glossentry}{name}{%
\def\@glo@name{#1}%
}
\define@key{glossentry}{description}{%
\def\@glo@desc{#1}%
}
\define@key{glossentry}{descriptionplural}{%
\def\@glo@descplural{#1}%
}
\define@key{glossentry}{sort}{%
\def\@glo@sort{#1}}
\define@key{glossentry}{text}{%
\def\@glo@text{#1}%
}
\define@key{glossentry}{plural}{%
\def\@glo@plural{#1}%
}
\define@key{glossentry}{first}{%
\def\@glo@first{#1}%
}
\define@key{glossentry}{firstplural}{%
\def\@glo@firstplural{#1}%
}
\newcommand*\@gls@default@value{\relax}
\define@key{glossentry}{symbol}{%
\def\@glo@symbol{#1}%
}
\define@key{glossentry}{symbolplural}{%
\def\@glo@symbolplural{#1}%
}
\define@key{glossentry}{type}{%
\def\@glo@type{#1}}
\define@key{glossentry}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%

```

```

    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
  {%
    \def\@glo@counter{#1}%
  }%
}
\define@key{glossentry}{see}{%
  \gls@set@xr@key{see}{\@glo@see}{#1}%
}
\newcommand*{\gls@set@xr@key}[3]{%
  \renewcommand*{\gls@xr@key}{#1}%
  \gls@checkseeallowed
  \def#2{#3}%
  \@glo@seeautonumberlist
}
\newcommand*{\gls@xr@key}{see}
\newcommand*{\gls@checkseeallowed}{%
  \@gls@see@noindex
}
\newcommand*{\gls@checkseeallowed@preambleonly}{%
  \GlossariesWarning
  {‘\gls@xr@key’ key doesn’t have any effect when used in the document
  environment. Move the definition to the preamble
  after \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@key{glossentry}{parent}{%
  \def\@glo@parent{#1}}
\define@choicekey{glossentry}{nonumberlist}{%
  [\gls@nonumberlist@val\gls@nonumberlist@nr]{true,false}[true]}%
  {%
    \ifcase\gls@nonumberlist@nr\relax
      \def\@glo@prefix{\glsnonextpages}%
      \@gls@savenonumberlist{true}%
    \else
      \def\@glo@prefix{\glsnextpages}%
      \@gls@savenonumberlist{false}%
    \fi
  }
\newcommand*{\gls@noidx@nonumberlist}[1]{%
  \cslet\glo@\@glo@label @prenumberlist}\@glo@prefix
}%
\newcommand{\glsnoidxprenumberlist}[1]{%
  \csuse{glo@\glsdetoklabel{#1}@prenumberlist}}
\newcommand*{\@gls@savenonumberlist}[1]{%
}
\newcommand*{\@gls@initnonumberlist}{%
}

```

```

\newcommand*{\@gls@storenonumberlist}[1]{%
\newcommand*{\@gls@enablesavenonumberlist}{%
  \renewcommand*{\@gls@initnonumberlist}{%
    \undef\@glo@nonumberlist
  }%
  \renewcommand*{\@gls@savenonumberlist}[1]{%
    \def\@glo@nonumberlist{##1}%
  }%
  \renewcommand*{\@gls@storenonumberlist}[1]{%
    \ifdef\@glo@nonumberlist
    {%
      \cslet{glo@glsdetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
    }%
    {}%
  }%
  \appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}
\define@key{glossentry}{user1}{%
  \def\@glo@useri{#1}%
}
\define@key{glossentry}{user2}{%
  \def\@glo@userii{#1}%
}
\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#1}%
}
\define@key{glossentry}{short}{%
  \def\@glo@short{#1}%
}
\define@key{glossentry}{shortplural}{%
  \def\@glo@shortpl{#1}%
}
\define@key{glossentry}{long}{%
  \def\@glo@long{#1}%
}
\define@key{glossentry}{longplural}{%
  \def\@glo@longpl{#1}%
}
\newcommand*{\@gls@name}{%
  \PackageError{glossaries}{name key required in
  \string\newglossaryentry\space for entry '\@glo@label'}{You

```

```

haven't specified the entry name}}
\newcommand*\@glsnodesc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '\@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}%
\newcommand*\@gls@missingnumberlist}[1]{%
  ??%
  \ifglssavenumberlist
  \GlossariesWarning{Missing number list for entry '#1'.
    Maybe makeglossaries + rerun required}%
  \else
  \PackageError{glossaries}%
  {Package option 'savenumberlist=true' required}%
  {%
    You must use the 'savenumberlist' package option
    to reference location lists.%
  }%
  \fi
}
\newcommand*\@gls@defaultsort{\@glo@name}
\newcount\gls@level
\newcommand*\@gls@noexpand@field}[3]{%
  \expandafter\global\expandafter
  \let\csname glo@#1@#2\endcsname#3%
}
\newcommand*\@gls@noexpand@fields}[4]{%
  \ifcsdef{gls@assign@#3@field}
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
    }%
    {%
      \csuse{gls@assign@#3@field}{#2}{#4}%
    }%
  }%
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \@gls@noexpand@field{#2}{#3}{\@gls@value}%
    }%
    {%
  }
}

```

```

        \@@gls@noexpand@field{#2}{#3}{#4}%
    }%
}
\newcommand{\@@gls@expand@field}[3]{%
\expandafter
\protected@xdef\csname glo@#1@#2\endcsname{#3}%
}
\newcommand{\@gls@expand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
{%
\@@gls@expand@field{#2}{#3}{#4}%
}%
{%
\csuse{gls@assign@#3@field}{#2}{#4}%
}%
}%
}%
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\@@gls@expand@field{#2}{#3}{#1}%
}%
{%
\@@gls@expand@field{#2}{#3}{#4}%
}%
}%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
\def\@gls@tmp{#1}%
\ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*\{gls@expandfields}{%
\let\gls@assign@field\@gls@expand@fields
}
\newcommand*\{gls@noexpandfields}{%
\let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
\glsdoifnoexists{#1}%

```

```

    {%
      \gls@defglossaryentry{#1}{#2}%
    }%
  }
\newcommand*\gls@defdocnewglossaryentry{%
  \let\newglossaryentry\new@glossaryentry
}
\newrobustcmd{\provideglossaryentry}[2]{%
  \ifglsentryexists{#1}%
  {%
    {%
      \gls@defglossaryentry{#1}{#2}%
    }%
  }
}
\@onlypreamble{\provideglossaryentry}
\newcommand*\gls@glsdefs@labellist{}
\newcommand*\gls@glsdefs@addlabel}[1]{\listxadd{\gls@glsdefs@labellist}{#1}}
\newcommand*\gls@docdef@labellist{}
\newcommand*\gls@docdef@addlabel}[1]{\listxadd{\gls@docdef@labellist}{#1}}
\newrobustcmd{\new@glossaryentry}[2]{%
  \ifundef\@gls@deffile
  {%
    \global\newwrite\@gls@deffile
    \immediate\openout\@gls@deffile=\jobname.glsdefs
  }%
  {%
    \ifglsentryexists{#1}%
    {%
      \xifinlist{#1}{\gls@docdef@labellist}%
      {%
        \GlossariesWarning{Entry ‘#1’ has already been defined in the
          document environment (ignoring new definition)}%
      }%
      {%
        \xifinlist{#1}{\gls@glsdefs@labellist}%
        {%
          \gls@defglossaryentry{#1}{#2}%
          \gls@docdef@addlabel{#1}%
          \@gls@writedef{#1}%
        }%
        {%
          \GlossariesWarning{Entry ‘#1’ has already been defined in
            the preamble (in which case remove the duplicate definition)
            or you have just updated to glossaries.sty
            v4.47 (in which case delete ‘\jobname.glsdefs’)}%
        }%
      }%
    }%
  }%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}

```

```

        \gls@docdef@addlabel{#1}%
        \@gls@writedef{#1}%
    }%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*\gls@begindocdefs{%
    \gls@enablesavenonumberlist
    \edef\@gls@restreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
    \makeatletter
    \InputIfFileExists{\jobname.glsdefs}{-}{-}%
    \@gls@restreat
    \undef\@gls@restreat
    \gls@defdocnewglossaryentry
}
\newcommand*\glsdefs@newdocentry}[2]{%
    \ifglsentryexists{#1}%
    {\GlossariesWarning{entry ‘#1’ has already been defined in the preamble
        (if you have moved the definition there from the document
        environment, delete \jobname.glsdefs)}}%
    {%
        \gls@glsdefs@addlabel{#1}%
        \gls@defglossaryentry{#1}{#2}%
    }%
}
\newcommand*\@gls@writedef}[1]{%
    \immediate\write\@gls@deffile
    {%
        \string\glsdefs@newdocentry{\glsdetoklabel{#1}}\glspercentchar^^J%
        \expandafter\@gobble\string{\glspercentchar%
    }%
    \@for\@gls@map:=\@gls@keymap\do
    {%
        \letcs\glo@value{glo@\glsdetoklabel{#1}}\expandafter\@secondoftwo\@gls@map}%
        \ifdef\glo@value
        {%
            \@onelevel@sanitize\glo@value
            \immediate\write\@gls@deffile
            {%
                \expandafter\@firstoftwo\@gls@map
                =\expandafter\@gobble\string{\glo@value\expandafter\@gobble\string\},%
                \glspercentchar
            }%
        }%
    }%
}
\gls@writedefhook
\immediate\write\@gls@deffile
{%
    \glspercentchar^^J%
}

```



```

    \expandafter\@gobble\string\}\glspercentchar^^J%
  }%
}
\newcommand*{\@gls@keymap}{%
  {name}{name},%
  {sort}{sortvalue},% unescaped sort value
  {type}{type},%
  {first}{first},%
  {firstplural}{firstpl},%
  {text}{text},%
  {plural}{plural},%
  {description}{desc},%
  {descriptionplural}{descplural},%
  {symbol}{symbol},%
  {symbolplural}{symbolplural},%
  {user1}{useri},%
  {user2}{userii},%
  {user3}{useriii},%
  {user4}{useriv},%
  {user5}{userv},%
  {user6}{uservi},%
  {long}{long},%
  {longplural}{longpl},%
  {short}{short},%
  {shortplural}{shortpl},%
  {counter}{counter},%
  {parent}{parent}%
}
\newcommand*{\@gls@fetchfield}[2]{%
  \edef\@gls@thisval{#2}%
  \@for\@gls@map:=\@gls@keymap\do{%
    \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
    \ifdefequal{\@this@key}{\@gls@thisval}%
    {%
      \edef#1{\expandafter\@secondoftwo\@gls@map}%
      \@endfortrue
    }%
  }%
}%
}
\newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
\newcommand*{\@sglsaddstoragekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
}%
}

```

```

\@glsaddstoragekey{#1}%
}
\newcommand*{\@glsaddstoragekey}[3]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{\@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
}%
{%
\PackageError{glossaries}{Key ‘#1’ already exists}{}%
}%
}
\newcommand*{\glsaddkey}{\@ifstar\@sglsaddkey\@glsaddkey}
\newcommand*{\@sglsaddkey}[1]{%
\key@ifundefined{glossentry}{#1}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname gls@assign@#1@field\endcsname}[2]{%
\@gls@expand@field{##1}{#1}{##2}%
}%
}%
{}%
\@glsaddkey{#1}%
}
\newcommand*{\@glsaddkey}[7]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{\@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
\newcommand*{#4}[1]{\@Glsentryfield{##1}{#1}}%
\ifcsdef{@gls@user@#1@}%
{%
\PackageError{glossaries}%
{Can’t define ‘\string#5’ as helper command
‘\expandafter\string\csname @gls@user@#1@\endcsname’ already exists}%
{}%
}%
{}%
}

```

```

\expandafter\newcommand\expandafter*\expandafter
  {\csname @gls@user@#1\endcsname}[2][]{%
    \new@ifnextchar[%
      {\csuse{@gls@user@#1@}{##1}{##2}}%
      {\csuse{@gls@user@#1@}{##1}{##2}[]}}%
\csdef{@gls@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{#3{##2}##3}%
}%
\newrobustcmd*{#5}{%
  \expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
}%
\ifcsdef{@Gls@user@#1@}{%
  {%
    \PackageError{glossaries}{%
      {Can't define '\string#6' as helper command
        '\expandafter\string\csname @Gls@user@#1\endcsname' already exists}}%
    }%
  }%
  {%
    \expandafter\newcommand\expandafter*\expandafter
      {\csname @Gls@user@#1\endcsname}[2][]{%
        \new@ifnextchar[%
          {\csuse{@Gls@user@#1@}{##1}{##2}}%
          {\csuse{@Gls@user@#1@}{##1}{##2}[]}}%
\csdef{@Gls@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{#4{##2}##3}%
}%
\newrobustcmd*{#6}{%
  \expandafter\@gls@hyp@opt\csname @Gls@user@#1\endcsname}%
}%
\ifcsdef{@GLS@user@#1@}{%
  {%
    \PackageError{glossaries}{%
      {Can't define '\string#7' as helper command
        '\expandafter\string\csname @GLS@user@#1\endcsname' already exists}}%
    }%
  }%
  {%
    \expandafter\newcommand\expandafter*\expandafter
      {\csname @GLS@user@#1\endcsname}[2][]{%
        \new@ifnextchar[%
          {\csuse{@GLS@user@#1@}{##1}{##2}}%
          {\csuse{@GLS@user@#1@}{##1}{##2}[]}}%
\csdef{@GLS@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{\glsuppercase{#3{##2}##3}}%
}%
\newrobustcmd*{#7}{%
  \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
}%
\glsmfuaddmap{#3}{#4}%

```

```

    \glsmfuaddmap{#5}{#6}%
    \glsmfublocker{#7}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#1’ already exists}{}%
  }%
}
\newrobustcmd{\glsfieldxdef}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \protected@csxdef{glo@\@glo@label @#2}{#3}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newrobustcmd{\glsfieldedef}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \protected@csedef{glo@\@glo@label @#2}{#3}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newrobustcmd{\glsfieldgdef}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newrobustcmd{\glsfieldddef}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%

```

```

\ifcsdef{glo@\@glo@label @#2}%
{%
  \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
}%
{%
  \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
}%
}%
}
\newrobustcmd{\glsfieldfetch}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \letcs#3{glo@\@glo@label @#2}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand*\glsunexpandedfieldvalue[2]{%
  \ifcsname glo@\glsdetoklabel{#1}@#2\endcsname
  \expandafter\expandonce\csname glo@\glsdetoklabel{#1}@#2\expandafter\endcsname
  \fi
}
\newrobustcmd{\ifglsfieldeq}[5]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newrobustcmd{\ifglsfielddefeq}[5]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \expandafter\ifdefstrequal\csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}

```

```

    }%
  }%
}
\newcommand{\ifglsfieldcseq}[5]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\glo@label @#2}%
    {%
      \ifcsstrequal{glo@\glo@label @#2}{#3}{#4}{#5}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand*{\glswritedefhook}{}
\newcommand*{\gls@assign@desc}[1]{%
  \gls@assign@field{#1}{desc}{\@glo@desc}%
  \gls@assign@field{\@glo@desc}{#1}{descplural}{\@glo@descplural}%
}
\newcommand{\longnewglossaryentry}[3]{%
  \glsdoifnoexists{#1}%
  {%
    \bgroup
    \let\@org@newglossaryentryprehook\@newglossaryentryprehook
    \long\def\@newglossaryentryprehook{%
      \long\def\@glo@desc{#3\leavevmode\unskip\nopostdesc}%
      \@org@newglossaryentryprehook
    }%
    \renewcommand*{\gls@assign@desc}[1]{%
      \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
      \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@desc}%
    }
    \gls@defglossaryentry{#1}{#2}%
  \egroup
}
}
\@onlypreamble{\longnewglossaryentry}
\newcommand{\longprovideglossaryentry}[3]{%
  \ifglsentryexists{#1}{%
    {\longnewglossaryentry{#1}{#2}{#3}}%
  }
}
\@onlypreamble{\longprovideglossaryentry}
\newcommand{\gls@defglossaryentry}[2]{%
  \let\GlsSetQuote\gls@nosetquote
  \protected@edef\glo@label{\glsdetoklabel{#1}}%
  \let\glslabel\@glo@label
  \let\@glo@name\@glsname
  \let\@glo@desc\@glsnodesc

```

```

\let\@glo@descplural\@gls@default@value
\let\@glo@type\@gls@default@value
\let\@glo@symbol\@gls@default@value
\let\@glo@symbolplural\@gls@default@value
\let\@glo@text\@gls@default@value
\let\@glo@plural\@gls@default@value
\let\@glo@first\@gls@default@value
\let\@glo@firstplural\@gls@default@value
\let\@glo@sort\@gls@default@value
\let\@glo@counter\@gls@default@value
\def\@glo@see{ }%
\def\@glo@parent{ }%
\def\@glo@prefix{ }%
\@gls@initnonumberlist
\def\@glo@useri{ }%
\def\@glo@userii{ }%
\def\@glo@useriii{ }%
\def\@glo@useriv{ }%
\def\@glo@userv{ }%
\def\@glo@uservi{ }%
\def\@glo@short{ }%
\def\@glo@shortpl{ }%
\def\@glo@long{ }%
\def\@glo@longpl{ }%
\@newglossaryentryprehook
\setkeys{glossentry}{#2}%
\ifundef\glsdefaulttype
{%
  \PackageError{glossaries}%
  {No default glossary type (have you used ‘nomain’ by mistake?)}%
  {If you use package option ‘nomain’ you must define
   a new glossary before you can define entries}%
}%
}%
\gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
\protected@edef\@glo@type{\glsentrytype{\@glo@label}}%
\ifcsundef{glo@list@\@glo@type}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘\@glo@type’ has not been defined}%
  {You need to define a new glossary type, before making entries
   in it}%
}%
\ifignoredglossary\@glo@type
{%
  \ifx\@glo@desc\@glsnodesc
    \let\@glo@desc\@empty
  \fi
}%

```

```

    {%
  }%
  \protected@edef\@glo@list@\csname glo@list@\@glo@type\endcsname}%
  \expandafter\protected@xdef\csname glo@list@\@glo@type\endcsname{%
    \@glo@list@\@glo@label},}%
}%
\gls@level=0\relax
\ifx\@glo@parent\@empty
  \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
\else
  \ifdefequal\@glo@label\@glo@parent%
    {%
      \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
      \def\@glo@parent{}%
      \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
    }%
    {%
      \ifglsentryexists{\@glo@parent}%
      {%
        \expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
          \@glo@parent}%
        \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
        \advance\gls@level by 1\relax
        \ifx\@glo@name\@gls@name
          \expandafter\let\expandafter\@glo@name
            \csname glo@\@glo@parent @name\endcsname
          \ifx\@glo@plural\@gls@default@value
            \expandafter\let\expandafter\@glo@plural
              \csname glo@\@glo@parent @plural\endcsname
          \fi
        \fi
      }%
    }%
    \PackageError{glossaries}%
    {%
      Invalid parent ‘\@glo@parent’
      for entry ‘\@glo@label’ - parent doesn’t exist%
    }%
    {%
      Parent entries must be defined before their children%
    }%
    \def\@glo@parent{}%
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
  }%
}%
\fi
\expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%
\gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
\letcs\@glo@sort{glo@\@glo@label @sortvalue}%
\gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%

```



```

\expandafter\gls@assign@field\expandafter
  {\csname glo@\glo@label @text\endcsname\glspluralsuffix}%
  {\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\glo@label @text\endcsname}%
  {\@glo@label}{first}{\@glo@first}%
\ifx\@glo@first\@gls@default@value
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\glo@label @plural\endcsname}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\else
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\glo@label @first\endcsname\glspluralsuffix}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\fi
\ifcsundef{@glo@type@\@glo@type @counter}%
{%
  \def\@glo@defaultcounter{\glscounter}%
}%
{%
  \letcs\@glo@defaultcounter{@glo@type@\@glo@type @counter}%
}%
\gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
\gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
\gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
\gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
\gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
\gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
\gls@assign@field{}{\@glo@label}{servi}{\@glo@servi}%
\gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
\gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
\gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
\gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%
\ifx\@glo@name\@gls@noname
  \@gls@noname
  \let\@glo@name\@gls@default@value
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%
\ifcsundef{glo@\glo@label @numberlist}%
{%
  \csxdef{glo@\glo@label @numberlist}{%
    \noexpand\@gls@missingnumberlist{\@glo@label}}%
}%
{%
  \@gls@storenonumberlist{\@glo@label}%
  \def\@glo@@desc{\@glo@first}%
  \ifx\@glo@desc\@glo@@desc
    \let\@glo@desc\@glo@first
  \fi
\ifx\@glo@desc\@gls@nodesc

```

```

    \@glsnodesc
    \let\@glsdesc\@gls@default@value
\fi
\gls@assign@desc{\@glo@label}%
\@gls@defsort{\@glo@type}{\@glo@label}%
\def\@glo@@symbol{\@glo@text}%
\ifx\@glo@symbol\@glo@@symbol
    \let\@glo@symbol\@glo@text
\fi
\gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
\expandafter
    \gls@assign@field\expandafter
        {csname glo@\@glo@label @symbol\endcsname}
        {\@glo@label}{symbolplural}{\@glo@symbolplural}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagfalse\endcsname{%
\noexpand\global
    \noexpand\let\expandafter\noexpand
        \csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagtrue\endcsname{%
\noexpand\global
    \noexpand\let\expandafter\noexpand
        \csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
}%
\csname glo@\@glo@label @flagfalse\endcsname
\@glo@autosee
\ifignoredglossary\@glo@type
{%
    \csdef{glo@\@glo@label @index}{}%
}
{%
    \do@glo@storeentry{\@glo@label}%
}%
\@newglossaryentry@defcounters
\@newglossaryentryposthook
}
\newcommand*{\@glo@autosee}{%
    \ifdefvoid\@glo@see{%
        {%
            \protected@edef\@do@glssee{%
                \noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
                \noexpand\expandafter\noexpand\@glssee\noexpand\@glo@list{\@glo@label}}%
            \@do@glssee
        }%
        \@glo@autoseehook
    }%
\newcommand*{\@glo@autoseehook}{%
\newcommand*{\@newglossaryentryprehook}{%
\newcommand*{\@newglossaryentryposthook}{%
\newcommand*{\@newglossaryentrydefcounters}{%

```

```

\newcommand*\glsmoveentry}[2]{%
  \protected@edef\glo@thislabel{\glsdetoklabel{#1}}%
  \protected@edef\glo@type{\csname glo@\glo@thislabel @type\endcsname}%
  \def\glo@list{,%}
  \forlslentries[\glo@type]{\glo@label}%
  {%
    \ifdefequal\glo@thislabel\glo@label
      {}{\protected@eappto\glo@list{\glo@label,}}%
    }%
  \cslet\glo@list@\glo@type{\glo@list}%
  \csdef\glo@\glo@thislabel @type}{#2}%
}
\ifglxindy
  \newcommand*\@glossaryentryfield{\string\glossentry}
\else
  \newcommand*\@glossaryentryfield{\string\glossentry}
\fi
\ifglxindy
  \newcommand*\@glossarysubentryfield{%
    \string\subglossentry}
\else
  \newcommand*\@glossarysubentryfield{%
    \string\subglossentry}
\fi
\newcommand{\glo@storeentry}[1]{%
  \protected@edef\glo@esclabel{#1}%
  \gls@checkmkidxchars\glo@esclabel
  \protected@edef\glo@sort{\csname glo@#1@sort\endcsname}%
  \gls@checkmkidxchars\glo@sort
  \gls@checkmkidxchars\glo@prefix
  \protected@edef\glo@parent{\csname glo@#1@parent\endcsname}%
  \ifglxindy
    \ifx\glo@parent\@empty
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        (\string\glo@sort\string" %
        \string\glo@prefix\@glossaryentryfield{\glo@esclabel}\string") %
      }%
    \else
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \csname glo@\glo@parent @index\endcsname
        (\string\glo@sort\string" %
        \string\glo@prefix\@glossarysubentryfield
        {\csname glo@#1@level\endcsname}{\glo@esclabel}\string") %
      }%
    \fi
  \else
    \ifx\glo@parent\@empty
      \@onelevel@sanitize\glo@prefix
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \glo@sort\gls@actualchar\glo@prefix

```

```

        \@glossaryentryfield{\@glo@esclabel}%
    }%
\else
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
\csname glo@\@glo@parent @index\endcsname\@gls@levelchar
\@glo@sort\@gls@actualchar\@glo@prefix
\@glossarysubentryfield
{\csname glo@#1@level\endcsname}\@glo@esclabel}%
}%
\fi
\fi
}
\newcommand*\@gls@if@measuring}[2]{%
\ifmeasuring@
#1%
\else
#2%
\fi
}
\let\gls@org@refstepcounter\refstepcounter
\let\gls@org@label\label
\newcommand*\gls@if@measuring}[2]{#2}
\AtBeginDocument{%
\@ifpackageloaded{amsmath}%
{\let\gls@if@measuring\@gls@if@measuring}%
{}}%
\let\gls@org@refstepcounter\refstepcounter
\let\gls@org@label\label
}
\newcommand*\gls@ifnotmeasuring[1]{\glsifmeasuring}{#1}}
\newcommand*\glsifmeasuring{\gls@if@measuring}
\newcommand*\gls@local@stepcounter}[1]{%
\ifcsdef{c@#1}%
{%
\epreto\gls@measuring@reset{\noexpand\setcounter{#1}{\number\value{#1}}}%
\expandafter\advance\csname c@#1\endcsname by \@ne
}%
{}}%
}
\newcommand*\gls@start@measuring}{%
\ifx\glsifmeasuring\@firstoftwo
\else
\let\gls@org@target\glstarget
\let\gls@org@link\glslink
\def\gls@measuring@reset{}%
\fi
\let\glsifmeasuring\@firstoftwo
\let\refstepcounter\gls@local@stepcounter
\let\label\@gobble
\let\glstarget\@secondoftwo

```

```

\let\@glslink\@secondoftwo
}
\newcommand{\gls@stop@measuring}{%
\renewcommand*\gls@ifmeasuring{\gls@if@measuring}%
\let\refstepcounter\gls@org@refstepcounter
\let\label\gls@org@label
\let\glstarget\gls@org@target
\let\@glslink\gls@org@link
\gls@measuring@reset
\def\gls@measuring@reset{%
}
\newcommand{\glsmeasureheight}[2]{%
\settoheight{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\newcommand{\glsmeasuredepth}[2]{%
\settodepth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\newcommand{\glsmeasurewidth}[2]{%
\settowidth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\def\@gls@patchtabularx#1\hbox#2#3!!{%
\def\TX@trial##1{#1\hbox{\gls@start@measuring#2\gls@stop@measuring}#3}%
}
\newcommand*\glspatchtabularx{%
\ifdef\TX@trial
{%
\expandafter\@gls@patchtabularx\TX@trial{##1}!!%
\let\glspatchtabularx\relax
}%
}%
}
\newcommand*\glsreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glsreset{#1}%
}%
}%
}
\newcommand*\glslocalreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glslocalreset{#1}%
}%
}%
}
\newcommand*\glsunset}[1]{%

```

```

\gls@ifnotmeasuring
{%
  \glsdoifexists{#1}%
  {%
    \@glsunset{#1}%
  }%
}%
}
\newcommand*\@glslocalunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glslocalunset{#1}%
    }%
  }%
}
\newcommand*\@glslocalunset{\@glslocalunset}
\newcommand*\@glslocalunset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
}
\newcommand*\@glsunset{\@glsunset}
\newcommand*\@glsunset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
}
\newcommand*\@glslocalreset{\@glslocalreset}
\newcommand*\@glslocalreset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
}
\newcommand*\@glsreset{\@glsreset}
\newcommand*\@glsreset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
}
\newcommand*\glsresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsreset{\@glsentry}%
  }%
}
\newcommand*\glslocalresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalreset{\@glsentry}%
  }%
}
\newcommand*\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsunset{\@glsentry}%
  }%
}

```

```

}
\newcommand*\glslocalunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalunset{\@glsentry}%
  }%
}
\newcommand*\@newglossaryentry@defcounters{%
  \csdef{glo@\@glo@label @currcount}{0}%
  \csdef{glo@\@glo@label @prevcount}{0}%
}
\newif\ifglsresetcurrcount
\glsresetcurrcountfalse
\newcommand*\glsenableentrycount{%
  \let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
  \renewcommand*\gls@defdocnewglossaryentry{%
    \renewcommand*\newglossaryentry[2]{%
      \PackageError{glossaries}{\string\newglossaryentry\space
        may only be used in the preamble when entry counting has
        been activated}{If you use \string\glsenableentrycount\space
        you must place all entry definitions in the preamble not in
        the document environment}%
    }%
  }%
}
\newcommand*\glsentrycurrcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
  {0}{\@gls@entry@field{##1}{currcount}}%
}
\newcommand*\glsentryprevcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
  {0}{\@gls@entry@field{##1}{prevcount}}%
}
\renewcommand*\@glsunset}[1]{%
  \@glsunset{##1}%
  \@gls@increment@currcount{##1}%
}
\renewcommand*\@glslocalunset}[1]{%
  \@glslocalunset{##1}%
  \@gls@local@increment@currcount{##1}%
}
\renewcommand*\@glsreset}[1]{%
  \@glsreset{##1}%
  \ifglsresetcurrcount\csgdef{glo@\glsdetoklabel{##1}@currcount}{0}\fi
}
\renewcommand*\@glslocalreset}[1]{%
  \@glslocalreset{##1}%
  \ifglsresetcurrcount\csdef{glo@\glsdetoklabel{##1}@currcount}{0}\fi
}
\def\@cgls@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax

```

```

        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \@gls@{##1}{##2}[##3]%
    \fi
}%
\def\@cglsp@##1##2[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \@cglsp@{##1}{##2}[##3]%
    \fi
}%
\def\@Gls@##1##2[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \@Gls@{##1}{##2}[##3]%
    \fi
}%
\def\@Glspl@##1##2[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \@Glspl@{##1}{##2}[##3]%
    \fi
}%
}
\AtEndDocument{\@gls@write@entrycounts}%
\renewcommand*{\@gls@entry@count}[2]{%
    \csdef{glo@\glsdetoklabel{##1}@prevcount}{##2}%
}%
\let\glsetenableentrycount\relax
}
\@onlypreamble\glsetenableentrycount
\newcommand*{\@gls@increment@currcount}[1]{%
    \csxdef{glo@\glsdetoklabel{##1}@currcount}{%
        \number\numexpr\glsetcurrcount{##1}+1}%
}
\newcommand*{\@gls@local@increment@currcount}[1]{%
    \csedef{glo@\glsdetoklabel{##1}@currcount}{%
        \number\numexpr\glsetcurrcount{##1}+1}%
}
\newcommand*{\@gls@write@entrycounts}{%
    \immediate\write\@auxout
        {\string\providecommand*{\string\@gls@entry@count}[2]{}}%
    \forallglsentries{\@glsentry}{%
        \ifglsused{\@glsentry}%

```



```

        {\immediate\write\@auxout
         {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}%
        }%
    }%
}
\newcommand*{\@gls@entry@count}[2]{%
\newrobustcmd*{\cgl}{\@gls@hyp@opt\@cgl}
\newcommand*{\@cgl}[2][{}]{%
    \new@ifnextchar[{\@cgl@{#1}{#2}}{\@cgl@{#1}{#2}[]}%
}
\def\@cgl@#1#2[#3]{%
    \GlossariesWarning{\string\cgl\space is defaulting to
        \string\gls\space since you haven't enabled entry counting}%
    \@cgl@{#1}{#2}[#3]%
}
\newcommand*{\cglformat}[2]{%
    \ifglshaslong{#1}{\glsentrylong{#1}}{\glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cGls}{\@gls@hyp@opt\@cGls}
\glsmfuaddmap{\cgl}{\cGls}
\newcommand*{\@cGls}[2][{}]{%
    \new@ifnextchar[{\@cGls@{#1}{#2}}{\@cGls@{#1}{#2}[]}%
}
\def\@cGls@#1#2[#3]{%
    \GlossariesWarning{\string\cGls\space is defaulting to
        \string\Gls\space since you haven't enabled entry counting}%
    \@Gls@{#1}{#2}[#3]%
}
\newcommand*{\cGlsformat}[2]{%
    \ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cglspl}{\@gls@hyp@opt\@cglspl}
\newcommand*{\@cglspl}[2][{}]{%
    \new@ifnextchar[{\@cglspl@{#1}{#2}}{\@cglspl@{#1}{#2}[]}%
}
\def\@cglspl@#1#2[#3]{%
    \GlossariesWarning{\string\cglspl\space is defaulting to
        \string\glspl\space since you haven't enabled entry counting}%
    \@glspl@{#1}{#2}[#3]%
}
\newcommand*{\cglsplformat}[2]{%
    \ifglshaslong{#1}{\glsentrylongpl{#1}}{\glsentryfirstplural{#1}}#2%
}
\newrobustcmd*{\cGlspl}{\@gls@hyp@opt\@cGlspl}
\glsmfuaddmap{\cglspl}{\cGlspl}
\newcommand*{\@cGlspl}[2][{}]{%
    \new@ifnextchar[{\@cGlspl@{#1}{#2}}{\@cGlspl@{#1}{#2}[]}%
}
\def\@cGlspl@#1#2[#3]{%
    \GlossariesWarning{\string\cGlspl\space is defaulting to

```

```

\string\Glspl\space since you haven't enabled entry counting}%
\@Glspl@{#1}{#2}[#3]%
}
\newcommand*\cGlsplformat}[2]{%
\ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
}
\newcommand*\loadglsentries}[2][\@gls@default]{%
\let\@gls@default\glsdefaulttype
\def\glsdefaulttype{#1}\input{#2}%
\let\glsdefaulttype\@gls@default
}
\@onlypreamble{\loadglsentries}
\newcommand*\glstextformat}[1]{#1}
\newcommand*\glsentryfmt{\glsgenentryfmt}
\newcommand*\glsgenentryfmt{%
\ifdefempty\glscustomtext
{%
\glsifplural
{%
\glscapscase
{%
\ifglsused\glslabel
{%
\glsentryplural{\glslabel}\glsinsert
}%
{%
\glsentryfirstplural{\glslabel}\glsinsert
}%
}%
{%
\ifglsused\glslabel
{%
\Glsentryplural{\glslabel}\glsinsert
}%
{%
\Glsentryfirstplural{\glslabel}\glsinsert
}%
}%
\ifglsused\glslabel
{%
\glsuppercase{\glsentryplural{\glslabel}\glsinsert}%
}%
{%
\glsuppercase{\glsentryfirstplural{\glslabel}\glsinsert}%
}%
}%
}%
\glscapscase

```

```

    {%
      \ifglused\glslabel
      {%
        \glsentrytext{\glslabel}\glsinsert
      }%
      {%
        \glsentryfirst{\glslabel}\glsinsert
      }%
    }%
    {%
      \ifglused\glslabel
      {%
        \Glsentrytext{\glslabel}\glsinsert
      }%
      {%
        \Glsentryfirst{\glslabel}\glsinsert
      }%
    }%
    {%
      \ifglused\glslabel
      {%
        \glsuppercase{\glsentrytext{\glslabel}\glsinsert}%
      }%
      {%
        \glsuppercase{\glsentryfirst{\glslabel}\glsinsert}%
      }%
    }%
  }%
}
\newcommand*{\glsnacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \acronymfont{\glsentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \glsuppercase
            {\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert}%
        }%
      }%
    }%
  }%
}

```



```

\newcommand*\genacrfullformat}[2]{%
  \glsentrylong{#1}#2\space
  (\protect\firstacronymfont{\glsentryshort{#1}})%
}
\newcommand*\Genacrfullformat}[2]{%
  \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}
\newcommand*\genplacrfullformat}[2]{%
  \glsentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glsentryshortpl{#1}})%
}
\newcommand*\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}
\define@key{glslink}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
  {%
    \def\@gls@counter{#1}%
  }%
}
\define@key{glslink}{format}{%
  \def\@glsnumberformat{#1}}
\newcommand*\@gls@glslink@hyper@update@hook{}
\define@boolkey{glslink}{hyper}[true]{\@gls@glslink@hyper@update@hook}
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\newcommand*\@gls@save@glslocal{%
  \let@if@org@KV@glslink@local@ifKV@glslink@local
}
\newcommand*\@gls@restore@glslocal{%
  \ifKV@glslink@local
    \let\@gls@do@glsunset@glslocalunset
  \else
    \let\@gls@do@glsunset@glsunset
  \fi
  \let@ifKV@glslink@local@if@org@KV@glslink@local
}
\newcommand*\@gls@do@glsunset}[1]{\glsunset{#1}}
\define@boolkey{glslink}{local}[true]{}
\@gls@save@glslocal
\newcommand*\glslinkvar}[3]{#1}

```

```

\newcommand*{\@gls@hyp@opt}[1]{%
  \let\glslinkvar\@firstofthree
  \let\@gls@hyp@opt@cs#1\relax
  \@ifstar{\s@gls@hyp@opt}%
  {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*{\s@gls@hyp@opt}[1] []{%
  \let\glslinkvar\@secondofthree
  \@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1] []{%
  \let\glslinkvar\@thirdofthree
  \@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
  \@gls@hyp@opt\@gls@@link
}
\newcommand*{\@gls@@link}[3] []{%
  \glsdoifexistsordo{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \@gls@link[#1]{#2}{#3}%
  }{%
    \glstextformat{#3}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Glslink}{%
  \@gls@hyp@opt\@Gls@@link
}
\newcommand*{\@Gls@@link}[3] []{%
  \@gls@@link[#1]{#2}{\glsentencecase{#3}}%
}
\glsmfuaddmap{\glslink}{\Glslink}
\newcommand*{\glspostlinkhook}{}
\newcommand*{\@gls@link@checkfirsthyper}{%
  \ifglsused{\glslabel}%
  {%
  }%
  {%
    \gls@checkisacronymlist\glstype
    \ifglshyperfirst
      \if@glsisacronymlist
        \ifglsacrfootnote
          \KV@glslink@hyperfalse
        \fi
      \fi
    \else
      \KV@glslink@hyperfalse
    \fi
  }%
  \glslinkcheckfirsthyperhook
}

```

```

}
\newcommand*\glslinkcheckfirsthyperhook{}
\newcommand*\glslinkpostsetkeys{}
\newcommand*\glsifhyperon[2]{\ifKV@glslink@hyper#1\else#2\fi}
\newcommand*\do@glsdisablehyperinlist{%
  \expandafter\DTLifinlist\expandafter{\gls\type}{\@gls@nohyperlist}%
  {\KV@glslink@hyperfalse}{}}
}
\newcommand*\@gls@setdefault@glslink@opts{}
\def\@gls@link[#1]#2#3{%
  \leavevmode
  \protected@edef\glslabel{\glsdetoklabel{#2}}%
  \def\@gls@link@opts{#1}%
  \let\@gls@link@label\glslabel
  \def\@glsnumberformat{glsnumberformat}%
  \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  \protected@edef\gls\type{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
  \@gls@save@glslocal
  \@gls@setdefault@glslink@opts
  \do@glsdisablehyperinlist
  \do@gls@link@checkfirsthyper
  \setkeys{glslink}{#1}%
  \glslinkpostsetkeys
  \@gls@saveentrycounter
  \@gls@setsort{\glslabel}%
  \glsencapwrcontent{\do@wrglossary{#2}}%
  \ifKV@glslink@hyper
    \@glslink{\glolinkprefix\glslabel}{\gls\textformat{#3}}%
  \else
    \glsdonohyperlink{\glolinkprefix\glslabel}{\gls\textformat{#3}}%
  \fi
  \let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
  \@gls@restore@glslocal
}
\newcommand*\glsencapwrcontent[1]{#1}
\newcommand*\glolinkprefix{glo:}
\def\glsentrycounter{\glscounter}%
\newcommand*\@gls@saveentrycounter{%
  \def\@gls@Hcounter{}%
  \ifthenelse{\equal{\@gls@counter}{equation}}%
  {
    \ifcsundef{xatlevel@}%
    {%
      \edef\theglsentrycounter{\expandafter\noexpand
        \csname the\@gls@counter\endcsname}%
    }%
  }%
  \ifx\xatlevel@\@empty
    \edef\theglsentrycounter{\expandafter\noexpand

```

```

        \csname the\@gls@counter\endcsname}%
\else
  \savecounters@
  \advance\c@equation by 1\relax
  \edef\theglentrycounter{\csname the\@gls@counter\endcsname}%
  \ifcsundef{theH\@gls@counter}%
  {%
    \def\@gls@Hcounter{\theglentrycounter}%
  }%
  {%
    \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
  }%
  \protected@edef\theHglentrycounter{\@gls@Hcounter}%
  \restorecounters@
\fi
}%
}%
{%
  \edef\theglentrycounter{\expandafter\noexpand
    \csname the\@gls@counter\endcsname}%
}%
\ifx\@gls@Hcounter\@empty
  \ifcsundef{theH\@gls@counter}%
  {%
    \def\theHglentrycounter{\theglentrycounter}%
  }%
  {%
    \protected@edef\theHglentrycounter{\expandafter\noexpand
      \csname theH\@gls@counter\endcsname}%
  }%
\fi
}
\def\@set@glo@numformat#1#2#3#4{%
  \expandafter\@glo@check@mkidxrangear#3\@nil
  \protected@edef#1{%
    \@glo@prefix setentrycounter[#4]{#2}%
    \expandafter\string\csname\@glo@suffix\endcsname
  }%
  \@gls@checkmkidxchars#1%
}
\def\@glo@check@mkidxrangear#1#2\@nil{%
\if#1(\relax
  \def\@glo@prefix{()}%
  \if\relax#2\relax
    \def\@glo@suffix{glsnumberformat}%
  \else
    \def\@glo@suffix{#2}%
  \fi
\else
  \if#1)\relax

```



```

\def\@glo@prefix{)}%
\if\relax#2\relax
\def\@glo@suffi{x{glsnumberformat}}%
\else
\def\@glo@suffi{x{#2}}%
\fi
\else
\def\@glo@prefix{ }\def\@glo@suffi{x{#1#2}}%
\fi
\fi}
\newcommand*\@gls@escbsdq}[1]{%
\def\@gls@checkedmkidx{}%
\let\gls@xdystring=#1\relax
\@onelevel@sanitize\gls@xdystring
\edef\do@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\@null}%
\do@gls@xdycheckbackslash
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\@null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\ifgls@esclocations
\@for\@gls@tmp:=\gls@protected@pagefmts\do
{%
\edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\ \expandonce\@gls@tmp}%
\@onelevel@sanitize\@gls@sanitized@tmp
\edef\gls@dostsubst{%
\noexpand\DTLsubstituteall\noexpand\gls@xdystring
{\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
}%
\gls@dostsubst
}%
\fi
\let#1=\gls@xdystring
}
\newcommand{\@gls@checkmkidxchars}[1]{%
\ifgls@xindy
\@gls@escbsdq{#1}%
\else
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkquote#1\@nil""\@null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkescquote#1\@nil""\@null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkescactual#1\@nil\?\?\@null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%

```

```

\expandafter\@gls@checkactual#1\@nil??\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkbar#1\@nil||\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkesbar#1\@nil|||\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checklevel#1\@nil!!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkesclevel#1\@nil!!\!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\edef\@do@gls@checkesc{%
  \noexpand\@gls@checkesc\expandonce{#1}\noexpand\@nil
  \@backslashchar\@backslashchar\noexpand\null
}%
\do@gls@checkesc
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\fi
}
\def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
\newtoks\@gls@tmpb
\def\@gls@checkquote#1"#2"#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
  \ifx\null#3\null
  \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
  \def\@@gls@checkquote{\relax}%
  \else
  \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
    \@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
  \def\@@gls@checkquote{\@gls@checkquote#3\null}%
  \fi
  \else
  \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
    \@gls@quotechar\@gls@quotechar}%
  \ifx\null#3\null
  \def\@@gls@checkquote{\@gls@checkquote#2""\null}%
  \else
  \def\@@gls@checkquote{\@gls@checkquote#2"#3\null}%
  \fi
  \fi
  \@@gls@checkquote
}
\def\@gls@checkescquote#1"#2"#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%

```

```

\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar
\@gls@quotechar\string"\@gls@quotechar}%
\def\@gls@checkescquote{\@gls@checkescquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar}%
\ifx\null#3\null
\def\@gls@checkescquote{\@gls@checkescquote#2\""\null}%
\else
\def\@gls@checkescquote{\@gls@checkescquote#2\##3\null}%
\fi
\fi
\@gls@checkescquote
}
\def\@gls@checkescactual#1\?#2\?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar
\@gls@quotechar\string"\@gls@actualchar}%
\def\@gls@checkescactual{\@gls@checkescactual#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar}%
\ifx\null#3\null
\def\@gls@checkescactual{\@gls@checkescactual#2\?\?\null}%
\else
\def\@gls@checkescactual{\@gls@checkescactual#2\?#3\null}%
\fi
\fi
\@gls@checkescactual
}
\def\@gls@checkescbar#1\|#2\|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null

```

```

\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkeschar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@encapchar
\@gls@quotechar\string"\@gls@encapchar}%
\def\@gls@checkeschar{\@gls@checkeschar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkeschar{\@gls@checkeschar#2\|\|\null}%
\else
\def\@gls@checkeschar{\@gls@checkeschar#2\|#3\null}%
\fi
\fi
\@gls@checkeschar
}
\def\@gls@checkeslevel#1\!#2\!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkeslevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@levelchar
\@gls@quotechar\string"\@gls@levelchar}%
\def\@gls@checkeslevel{\@gls@checkeslevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checkeslevel{\@gls@checkeslevel#2\!\!\null}%
\else
\def\@gls@checkeslevel{\@gls@checkeslevel#2\!#3\null}%
\fi
\fi
\@gls@checkeslevel
}
\def\@gls@checkbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%

```

```

\def\@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@gls@checkbar{\@gls@checkbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkbar{\@gls@checkbar#2|\null}%
\else
\def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}
\def\@gls@checklevel#1!#2!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checklevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
\def\@gls@checklevel{\@gls@checklevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checklevel{\@gls@checklevel#2!\null}%
\else
\def\@gls@checklevel{\@gls@checklevel#2|#3\null}%
\fi
\fi
\@gls@checklevel
}
\def\@gls@checkactual#1?#2?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%

```

```

        \def\@gls@checkactual{\@gls@checkactual#3\null}%
    \fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar}%
\ifx\null#3\null
\def\@gls@checkactual{\@gls@checkactual#2??\null}%
\else
\def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
\fi
\fi
\@gls@checkactual
}
\bgroup
\catcode'\0\relax
\catcode'\12\relax
|gdef|\@gls@checkesc#1\#2\#3\null{%
|\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
|\toks@={#1}%
\ifx\null#2\null
|\ifx\null#3\null
|\edef|\@gls@checkedmkidx{\the|\@gls@tmpb|\the|\toks@}%
|\def|\@gls@checkesc{\relax}%
|\else
|\edef|\@gls@checkedmkidx{\the|\@gls@tmpb|\the|\toks@
|\@gls@quotechar|\backslashchar|\@gls@quotechar|\backslashchar}%
|\def|\@gls@checkesc{\@gls@checkesc#3\null}%
|\fi
|\else
|\edef|\@gls@checkedmkidx{\the|\@gls@tmpb|\the|\toks@
|\@gls@quotechar|\backslashchar}%
|\ifx\null#3\null
|\def|\@gls@checkesc{\@gls@checkesc#2\\\null}%
|\else
|\def|\@gls@checkesc{\@gls@checkesc#2\#3\null}%
|\fi
|\fi
|\@gls@checkesc
}
|egroup
\def\@gls@xdycheckquote#1"#2"#3\null{%
|\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
|\toks@={#1}%
\ifx\null#2\null
|\ifx\null#3\null
|\edef|\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
|\def\@gls@xdycheckquote{\relax}%
|\else
|\edef|\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
|\string"\string"}%

```

```

        \def\@@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
    \fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\string\}%
\ifx\null#3\null
\def\@@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
\else
\def\@@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
\fi
\fi
\fi
\@@gls@xdycheckquote
}
\edef\def@gls@xdycheckbackslash{%
\noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
##2\@backslashchar##3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter
{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={##1}%
\noexpand\ifx\noexpand\null##2\noexpand\null
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@@gls@xdycheckbackslash{\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
\noexpand\def\noexpand\@@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar}%
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\def\noexpand\@@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
\@backslashchar\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
##3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@@gls@xdycheckbackslash
}%
}
\def@gls@xdycheckbackslash

```

```

\newlength\gls@tmplen
\newcommand*\glsdohypertarget}[2]{%
  \glsdoshowtarget{#1}%
  {%
    \glsmeasureheight{\gls@tmplen}{#2}%
    \raisebox{\gls@tmplen}{\hypertarget{#1}{}}%
    #2%
  }%
}
\newcommand*\glsdohyperlink}[2]{%
  \glsdoshowtarget{#1}{\hyperlink{#1}{#2}}%
}
\newcommand*\glsdonohyperlink}[2]{#2}
\ifcsundef{hyperlink}%
{%
  \let\@glslink\glsdonohyperlink
}%
{%
  \let\@glslink\glsdohyperlink
}
\ifcsundef{hypertarget}%
{%
  \let\@glstarget\@secondoftwo
}%
{%
  \let\@glstarget\glsdohypertarget
}
\newcommand{\glsdisablehyper}{%
  \KV@glslink@hyperfalse
  \let\@glslink\glsdonohyperlink
  \let\@glstarget\@secondoftwo
}
\newcommand{\glsenablehyper}{%
  \KV@glslink@hypertrue
  \let\@glslink\glsdohyperlink
  \let\@glstarget\glsdohypertarget
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\newrobustcmd*\gls{\@gls@hyp@opt\@gls}
\newcommand*\@gls}[2][{}]{%
  \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2}[]}%
}
\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\gls@ifplural\@secondoftwo
    \let\gls@scapscase\@firstofthree
    \let\gls@customtext\@empty
  }
}

```



```

\def\glsinsert{#3}%
\def\glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
\glsmfuaddmap{\gls}{\Gls}
\newcommand*{\@Gls}[2][{}]{%
\new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2}[]]}%
}
\def\@Gls@#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@secondoftwo
\let\gls@caps@case\@secondofthree
\let\gls@custom@text\@empty
\def\glsinsert{#3}%
\def\glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}
\glsmfublocker{\GLS}
\newcommand*{\@GLS}[2][{}]{%
\new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2}[]]}%
}
\def\@GLS@#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@secondoftwo
\let\gls@caps@case\@thirdofthree
\let\gls@custom@text\@empty
\def\glsinsert{#3}%
\def\glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\glspl}{\@gls@hyp@opt\@glspl}
\newcommand*{\@glspl}[2][{}]{%
\new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2}[]]}%
}
\def\@glspl@#1#2[#3]{%

```

```

\glsdoifexists{#2}%
{%
  \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
  \let\glsifplural\@firstoftwo
  \let\glsescapscase\@firstofthree
  \let\glscustomtext\@empty
  \def\glsinsert{#3}%
  \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
  \@gls@link[#1]{#2}{\@glo@text}%
  \@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
\glsmfuaddmap{\glspl}{\Glspl}
\newcommand*{\@Glspl}[2][{}]{%
  \new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2}[]}%
}
\def\@Glspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glsescapscase\@secondofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \@gls@do@glsunset{#2}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLSp1}{\@gls@hyp@opt\@GLSp1}
\glsmfublocker{\GLSp1}
\newcommand*{\@GLSp1}[2][{}]{%
  \new@ifnextchar[{\@GLSp1@{#1}{#2}}{\@GLSp1@{#1}{#2}[]}%
}
\def\@GLSp1@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glsescapscase\@thirdofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \@gls@do@glsunset{#2}%
  }%
  \glspostlinkhook
}

```

```

}
\newrobustcmd*{\glsdisp}{\@gls@hyp@opt\@glsdisp}
\newcommand*{\@glsdisp}[3][{}]{%
  \glsdoifexists{#2}{%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\gls@scaps\@firstofthree
    \def\gls@customtext{#3}%
    \def\gls@insert{}%
    \def\@glo@text{\csname gls@\gls@type @entryfmt@endcsname}%
    \@gls@link{#1}{#2}{\@glo@text}%
    \@gls@do@gls@unset{#2}%
  }%
  \gls@postlinkhook
}
\newrobustcmd*{\Glsdisp}{\@gls@hyp@opt\@Glsdisp}
\newcommand*{\@Glsdisp}[3][{}]{%
  \@glsdisp{#1}{#2}{\gls@sentencecase{#3}}%
}
\gls@fu@addmap{\glsdisp}{\Glsdisp}
\newcommand*{\@gls@link@nocheckfirsthyper}{}
\newcommand{\@gls@field@link}[3]{%
  \glsdoifexists{#2}{%
    {%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
      \@gls@link{#1}{#2}{#3}%
    }%
  }%
  \gls@postlinkhook
}
\newrobustcmd*{\gls@text}{\@gls@hyp@opt\@gls@text}
\newcommand*{\@gls@text}[2][{}]{%
  \new@ifnextchar[{\@gls@text@{#1}{#2}}{\@gls@text@{#1}{#2}[]]}
\def\@gls@text@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@entrytext{#2}#3}%
}
\newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
\gls@fu@blocker{\GLStext}
\newcommand*{\@GLStext}[2][{}]{%
  \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2}[]]}
\def\@GLStext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@supercase{\gls@entrytext{#2}#3}}%
}
\newrobustcmd*{\Gls@text}{\@gls@hyp@opt\@Gls@text}
\gls@fu@addmap{\gls@text}{\Gls@text}
\newcommand*{\@Gls@text}[2][{}]{%
  \new@ifnextchar[{\@Gls@text@{#1}{#2}}{\@Gls@text@{#1}{#2}[]]}
\def\@Gls@text@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Gls@entrytext{#2}#3}%
}
\newrobustcmd*{\gls@first}{\@gls@hyp@opt\@gls@first}

```

```

\newcommand*{\@glsfirst}[2] [] {%
  \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2} []}]
\def\@glsfirst@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
}
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
\glsmfuaddmap{\glsfirst}{\Glsfirst}
\newcommand*{\@Glsfirst}[2] [] {%
  \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2} []}]
\def\@Glsfirst@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
\glsmfublocker{\GLSfirst}
\newcommand*{\@GLSfirst}[2] [] {%
  \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2} []}]
\def\@GLSfirst@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentryfirst{#2}#3}%
}
\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
\newcommand*{\@glsplural}[2] [] {%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} []}]
\def\@glsplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\glsmfuaddmap{\glsplural}{\Glsplural}
\newcommand*{\@Glsplural}[2] [] {%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} []}]
\def\@Glsplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
\newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
\glsmfublocker{\GLSplural}
\newcommand*{\@GLSplural}[2] [] {%
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} []}]
\def\@GLSplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{\@glsfirstplural}[2] [] {%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} []}]
\def\@glsfirstplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
\glsmfuaddmap{\glsfirstplural}{\Glsfirstplural}
\newcommand*{\@Glsfirstplural}[2] [] {%
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2} []}]
\def\@Glsfirstplural@#1#2[#3] {%

```

```

    \@gls@field@link{#1}{#2}{\glstentryfirstplural{#2}#3}%
}
\newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
\glsmfublocker{\GLSfirstplural}
\newcommand*{\@GLSfirstplural}[2] [] {%
    \new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} []}]
\def\@GLSfirstplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glssupercase{\glstentryfirstplural{#2}#3}}%
}
\newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
\newcommand*{\@glsname}[2] [] {%
    \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2} []}]
\def\@glsname@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glstentryname{#2}#3}%
}
\newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}
\glsmfuaddmap{\glsname}{\Glsname}
\newcommand*{\@Glsname}[2] [] {%
    \new@ifnextchar[{\@Glsname@{#1}{#2}}{\@Glsname@{#1}{#2} []}]
\def\@Glsname@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glstentryname{#2}#3}%
}
\newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}
\glsmfublocker{\GLSname}
\newcommand*{\@GLSname}[2] [] {%
    \new@ifnextchar[{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}]
\def\@GLSname@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glssupercase{\glstentryname{#2}#3}}%
}
\newrobustcmd*{\glsdesc}{\@gls@hyp@opt\@glsdesc}
\newcommand*{\@glsdesc}[2] [] {%
    \new@ifnextchar[{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2} []}]
\def\@glsdesc@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glstentrydesc{#2}#3}%
}
\newrobustcmd*{\Glsdesc}{\@gls@hyp@opt\@Glsdesc}
\glsmfuaddmap{\glsdesc}{\Glsdesc}
\newcommand*{\@Glsdesc}[2] [] {%
    \new@ifnextchar[{\@Glsdesc@{#1}{#2}}{\@Glsdesc@{#1}{#2} []}]
\def\@Glsdesc@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glstentrydesc{#2}#3}%
}
\newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\glsmfublocker{\GLSdesc}
\newcommand*{\@GLSdesc}[2] [] {%
    \new@ifnextchar[{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}]
\def\@GLSdesc@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glssupercase{\glstentrydesc{#2}#3}}%
}
\newrobustcmd*{\glsdescplural}{\@gls@hyp@opt\@glsdescplural}

```

```

\newcommand*{\@glsdescplural}[2] [] {%
  \new@ifnextchar[{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}]
\def\@glsdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\Glsdescplural}{\@gls@hyp@opt\@Glsdescplural}
\glsmfuaddmap{\glsdescplural}{\Glsdescplural}
\newcommand*{\@Glsdescplural}[2] [] {%
  \new@ifnextchar[{\@Glsdescplural@{#1}{#2}}{\@Glsdescplural@{#1}{#2} []}]
\def\@Glsdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\glsmfublocker{\GLSdescplural}
\newcommand*{\@GLSdescplural}[2] [] {%
  \new@ifnextchar[{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}]
\def\@GLSdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentrydescplural{#2}#3}}%
}
\newrobustcmd*{\glsymbol}{\@gls@hyp@opt\@glsymbol}
\newcommand*{\@glsymbol}[2] [] {%
  \new@ifnextchar[{\@glsymbol@{#1}{#2}}{\@glsymbol@{#1}{#2} []}]
\def\@glsymbol@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\GLSymbol}{\@gls@hyp@opt\@GLSymbol}
\glsmfublocker{\GLSymbol}
\newcommand*{\@GLSymbol}[2] [] {%
  \new@ifnextchar[{\@GLSymbol@{#1}{#2}}{\@GLSymbol@{#1}{#2} []}]
\def\@GLSymbol@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentrysymbol{#2}#3}}%
}
\newrobustcmd*{\glsymbolplural}{\@gls@hyp@opt\@glsymbolplural}
\newcommand*{\@glsymbolplural}[2] [] {%
  \new@ifnextchar[{\@glsymbolplural@{#1}{#2}}{\@glsymbolplural@{#1}{#2} []}]
\def\@glsymbolplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSymbolplural}{\@gls@hyp@opt\@GLSymbolplural}
\glsmfuaddmap{\glsymbolplural}{\GLSymbolplural}
\newcommand*{\@GLSymbolplural}[2] [] {%
  \new@ifnextchar[{\@GLSymbolplural@{#1}{#2}}{\@GLSymbolplural@{#1}{#2} []}]
\def\@GLSymbolplural@#1#2[#3] {%

```

```

    \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
\glsmfublocker{\GLSsymbolplural}
\newcommand*{\@GLSsymbolplural}[2] [] {%
  \new@ifnextchar[{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2} []}]
\def\@GLSsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrysymbolplural{#2}#3}}%
}
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
\newcommand*{\@glsuseri}[2] [] {%
  \new@ifnextchar[{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2} []}]
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuseri{#2}#3}%
}
\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
\glsmfuaddmap{\glsuseri}{\Glsuseri}
\newcommand*{\@Glsuseri}[2] [] {%
  \new@ifnextchar[{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2} []}]
\def\@Glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\glsmfublocker{\GLSuseri}
\newcommand*{\@GLSuseri}[2] [] {%
  \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2} []}]
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseri{#2}#3}}%
}
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
\newcommand*{\@glsuserii}[2] [] {%
  \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2} []}]
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserii{#2}#3}%
}
\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
\glsmfuaddmap{\glsuserii}{\Glsuserii}
\newcommand*{\@Glsuserii}[2] [] {%
  \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2} []}]
\def\@Glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\GLSuserii}{\@gls@hyp@opt\@GLSuserii}
\glsmfublocker{\GLSuserii}
\newcommand*{\@GLSuserii}[2] [] {%
  \new@ifnextchar[{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2} []}]
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuserii{#2}#3}}%
}
\newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}

```

```

\newcommand*{\@glsuseriii}[2] [] {%
  \new@ifnextchar[{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2} []}]
\def\@glsuseriii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
\glsmfuaddmap{\glsuseriii}{\Glsuseriii}
\newcommand*{\@Glsuseriii}[2] [] {%
  \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2} []}]
\def\@Glsuseriii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\GLSuseriii}{\@gls@hyp@opt\@GLSuseriii}
\glsmfublocker{\GLSuseriii}
\newcommand*{\@GLSuseriii}[2] [] {%
  \new@ifnextchar[{\@GLSuseriii@{#1}{#2}}{\@GLSuseriii@{#1}{#2} []}]
\def\@GLSuseriii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseriii{#2}#3}}%
}
\newrobustcmd*{\glsuseriv}{\@gls@hyp@opt\@glsuseriv}
\newcommand*{\@glsuseriv}[2] [] {%
  \new@ifnextchar[{\@glsuseriv@{#1}{#2}}{\@glsuseriv@{#1}{#2} []}]
\def\@glsuseriv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
\glsmfuaddmap{\glsuseriv}{\Glsuseriv}
\newcommand*{\@Glsuseriv}[2] [] {%
  \new@ifnextchar[{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2} []}]
\def\@Glsuseriv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\GLSuseriv}{\@gls@hyp@opt\@GLSuseriv}
\glsmfublocker{\GLSuseriv}
\newcommand*{\@GLSuseriv}[2] [] {%
  \new@ifnextchar[{\@GLSuseriv@{#1}{#2}}{\@GLSuseriv@{#1}{#2} []}]
\def\@GLSuseriv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseriv{#2}#3}}%
}
\newrobustcmd*{\glsuserv}{\@gls@hyp@opt\@glsuserv}
\newcommand*{\@glsuserv}[2] [] {%
  \new@ifnextchar[{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2} []}]
\def\@glsuserv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\Glsuserv}{\@gls@hyp@opt\@Glsuserv}
\glsmfuaddmap{\glsuserv}{\Glsuserv}
\newcommand*{\@Glsuserv}[2] [] {%
  \new@ifnextchar[{\@Glsuserv@{#1}{#2}}{\@Glsuserv@{#1}{#2} []}]
\def\@Glsuserv@#1#2[#3] {%

```



```

    \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
\glsmfublocker{\GLSuserv}
\newcommand*{\@GLSuserv}[2][{}]{%
\new@ifnextchar[{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2}}]}
\def\@GLSuserv@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
\newcommand*{\@glsuservi}[2][{}]{%
    \new@ifnextchar[{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2}}]}
\def\@glsuservi@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
\newrobustcmd*{\Glsuservi}{\@gls@hyp@opt\@Glsuservi}
\glsmfuaddmap{\glsuservi}{\Glsuservi}
\newcommand*{\@Glsuservi}[2][{}]{%
    \new@ifnextchar[{\@Glsuservi@{#1}{#2}}{\@Glsuservi@{#1}{#2}}]}
\def\@Glsuservi@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryuservi{#2}#3}%
}
\newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
\glsmfublocker{\GLSuservi}
\newcommand*{\@GLSuservi}[2][{}]{%
    \new@ifnextchar[{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2}}]}
\def\@GLSuservi@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuservi{#2}#3}%
}
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\@ns@acrshort}
\newcommand*{\@ns@acrshort}[2][{}]{%
    \new@ifnextchar[{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2}}]}
}
\def\@acrshort#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \let\glsifplural\@secondoftwo
        \let\glscapscase\@firstofthree
        \let\glsinsert\@empty
        \def\glscustomtext{%
            \acronymfont{\glsentryshort{#2}}#3%
        }%
        \@gls@link[#1]{#2}{\csname gls@\gls@type @entryfmt\endcsname}%
    }%
    \glspostlinkhook
}
\newrobustcmd*{\Acrshort}{\@gls@hyp@opt\@ns@Acrshort}
\glsmfuaddmap{\Acrshort}{\Acrshort}
\newcommand*{\@ns@Acrshort}[2][{}]{%

```

```

\new@ifnextchar [{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2} []}%
}
\def\@Acrshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@secondoftwo
\let\glscapscase\@secondofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\Glsentryshort{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\ACRshort}{\@gl@hyp@opt\@ns@ACRshort}
\glsmfublocker{\ACRshort}
\newcommand*{\ns@ACRshort}[2] []{%
\new@ifnextchar [{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}%
}
\def\@ACRshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\glsupercase{\acronymfont{\glsentryshort{#2}}#3}%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\acrshortpl}{\@gl@hyp@opt\@ns@acrshortpl}
\newcommand*{\ns@acrshortpl}[2] []{%
\new@ifnextchar [{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} []}%
}
\def\@acrshortpl#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%

```

```

        \acronymfont{\glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls@type @entryfmt\endcsname}%
}
\glspostlinkhook
}
\newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\ns@Acrshortpl}
\glsmfuaddmap{\Acrshortpl}{\Acrshortpl}
\newcommand*{\ns@Acrshortpl}[2] []{%
    \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2} []}%
}
\def\@Acrshortpl#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \def\glslabel{#2}%
        \let\glsifplural\@firstoftwo
        \let\gls@scaps\@secondofthree
        \let\glsinsert\@empty
        \def\gls@customtext{%
            \acronymfont{\Glsentryshortpl{#2}}#3%
        }%
        \@gls@link[#1]{#2}{\csname gls@\gls@type @entryfmt\endcsname}%
    }%
    \glspostlinkhook
}
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\ns@ACRshortpl}
\glsmfublocker{\ACRshortpl}
\newcommand*{\ns@ACRshortpl}[2] []{%
    \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2} []}%
}
\def\@ACRshortpl#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \def\glslabel{#2}%
        \let\glsifplural\@firstoftwo
        \let\gls@scaps\@thirdofthree
        \let\glsinsert\@empty
        \def\gls@customtext{%
            \glsuppercase{\acronymfont{\glsentryshortpl{#2}}#3}%
        }%
        \@gls@link[#1]{#2}{\csname gls@\gls@type @entryfmt\endcsname}%
    }%
    \glspostlinkhook
}
\newrobustcmd*{\acrlong}{\@gls@hyp@opt\ns@acrlong}
\newcommand*{\ns@acrlong}[2] []{%
    \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2} []}%
}

```

```

\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@firstofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \glsentrylong{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \gls post link hook
}
\newrobustcmd*{\Acrlong}{\@gls@hyp@opt\@ns@Acrlong}
\glsmfuaddmap{\Acrlong}{\Acrlong}
\newcommand*{\ns@Acrlong}[2][]{%
  \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2}[]}%
}
\def\@Acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@secondofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \Glsentrylong{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \gls post link hook
}
\newrobustcmd*{\ACrlong}{\@gls@hyp@opt\@ns@ACrlong}
\glsmfublocker{\ACrlong}
\newcommand*{\ns@ACrlong}[2][]{%
  \new@ifnextchar[{\@ACrlong{#1}{#2}}{\@ACrlong{#1}{#2}[]}%
}
\def\@ACrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@thirdofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \glsuppercase{\glsentrylong{#2}#3}%
    }%
  }%
}

```

```

    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\ns@acrlongpl}
\newcommand*{\ns@acrlongpl}[2][]{%
  \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2} []}%
}
\def\@acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\ns@Acrlongpl}
\glsmfuaddmap{\Acrlongpl}{\Acrlongpl}
\newcommand*{\ns@Acrlongpl}[2][]{%
  \new@ifnextchar[{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2} []}%
}
\def\@Acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glsapsaps\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\ns@ACRlongpl}
\glsmfublocker{\ACRlongpl}
\newcommand*{\ns@ACRlongpl}[2][]{%
  \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2} []}%
}
\def\@ACRlongpl#1#2[#3]{%

```

```

\glsdoifexists{#2}%
{%
  \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
  \def\glslabel{#2}%
  \let\glsifplural\@firstoftwo
  \let\gls caps case\@thirdofthree
  \let\glsinsert\@empty
  \def\gls custom text{%
    \glsupercase{\glsentrylongpl{#2}#3}%
  }%
  \@gls@link[#1]{#2}{\csname gls@\gls type @entryfmt\endcsname}%
}%
\gls post link hook
}
\newcommand*{\@gls@entry@field}[2]{%
  \csuse{glo@\glsdetoklabel{#1}@#2}%
}
\newcommand*{\glsletentryfield}[3]{%
  \letcs{#1}{glo@\glsdetoklabel{#2}@#3}%
}
\newrobustcmd*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@#2}%
    \ifdef\@glo@text
    {%
      \expandafter\gls sentence case\expandafter{\@glo@text}%
    }%
    {%
      ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
        entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
        label and the field name}%
    }%
  }%
  }%
  }%
  }%
}
\newcommand{\@Glsentryfield}[2]{%
  \gls tex or pdf string{\@Gls@entry@field{#1}{#2}}%
  {\MFUsentencecase{\@gls@entry@field{#1}{#2}}}%
}
\ifdef\tex or pdf string
{\newcommand{\gls tex or pdf string}{\tex or pdf string}}
{\newcommand{\gls tex or pdf string}[2]{#1}}
\newcommand*{\glsentryname}[1]{\@gls@entry@field{#1}{name}}
\newcommand*{\Glsentryname}[1]{%
  \gls tex or pdf string{\protect\@Gls@entryname{#1}}%
  {\MFUsentencecase{\@gls@entry@field{#1}{name}}}%
}

```

```

\glsmfuaddmap{\glsentryname}{\Glsentryname}
\newcommand*{\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
\newrobustcmd*{\@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \protected@edef\@glo@text{\@Gls@entry@field{#1}{name}}%
    \expandafter\glsentencecase\expandafter{\@glo@text}%
  }%
  {%
    \@Gls@entry@field{#1}{name}%
  }%
}
\newcommand*{\@org@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo\@glsdetoklabel{#1}@name}%
    \expandafter\@gls@getbody\@glo@text{}\@nil
    \expandafter\ifx\@gls@body\glsentrylong\relax
    \expandafter\Glsentrylong\@gls@rest
  \else
    \expandafter\ifx\@gls@body\glsentryshort\relax
    \expandafter\Glsentryshort\@gls@rest
  \else
    \expandafter\ifx\@gls@body\acronymfont\relax
    {%
      \let\glsentryshort\Glsentryshort
      \@glo@text
    }%
  \else
    \expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
    {%
      \let\glsentryshort\Glsentryshort
      \@glo@text
    }%
  \else
    \expandafter\glsentencecase\expandafter{\@glo@text}%
  \fi
  \fi
  \fi
  \fi
  }%
  {%
    \@Gls@entry@field{#1}{name}%
  }%
}
\newcommand*{\glsentrydesc}[1]{\@Gls@entry@field{#1}{desc}}
\newcommand*{\Glsentrydesc}[1]{\@Gls@entryfield{#1}{desc}}
\glsmfuaddmap{\glsentrydesc}{\Glsentrydesc}

```

```

\newcommand*\glsentrydescplural}[1]{%
  \@gls@entry@field{#1}{descplural}%
}
\newcommand*\Glsentrydescplural}[1]{%
  \@Glsentryfield{#1}{descplural}%
}
\glsmfuaddmap{\glsentrydescplural}{\Glsentrydescplural}
\newcommand*\glsentrytext}[1]{\@gls@entry@field{#1}{text}}
\newcommand*\Glsentrytext}[1]{%
  \@Glsentryfield{#1}{text}%
}
\glsmfuaddmap{\glsentrytext}{\Glsentrytext}
\newcommand*\glsentryplural}[1]{%
  \@gls@entry@field{#1}{plural}%
}
\newcommand*\Glsentryplural}[1]{%
  \@Glsentryfield{#1}{plural}%
}
\glsmfuaddmap{\glsentryplural}{\Glsentryplural}
\newcommand*\glsentrysymbol}[1]{%
  \@gls@entry@field{#1}{symbol}%
}
\newcommand*\Glsentrysymbol}[1]{%
  \@Glsentryfield{#1}{symbol}%
}
\glsmfuaddmap{\glsentrysymbol}{\Glsentrysymbol}
\newcommand*\glsentrysymbolplural}[1]{%
  \@gls@entry@field{#1}{symbolplural}%
}
\newcommand*\Glsentrysymbolplural}[1]{%
  \@Glsentryfield{#1}{symbolplural}%
}
\glsmfuaddmap{\glsentrysymbolplural}{\Glsentrysymbolplural}
\newcommand*\glsentryfirst}[1]{%
  \@gls@entry@field{#1}{first}%
}
\newcommand*\Glsentryfirst}[1]{%
  \@Glsentryfield{#1}{first}%
}
\glsmfuaddmap{\glsentryfirst}{\Glsentryfirst}
\newcommand*\glsentryfirstplural}[1]{%
  \@gls@entry@field{#1}{firstpl}%
}
\newcommand*\Glsentryfirstplural}[1]{%
  \@Glsentryfield{#1}{firstpl}%
}
\glsmfuaddmap{\glsentryfirstplural}{\Glsentryfirstplural}
\newcommand*\glscapitalisewords}[1]{%
  \capitalisewords{#1}%
}
}

```



```

\newcommand*\glsentrytitlecase}[2]{%
  \glstexorpdfstring
  {\@glsentrytitlecase{#1}{#2}}%
  {\MFUsentencecase{\@gls@entry@field{#1}{#2}}}%
}
\newrobustcmd*\@glsentrytitlecase}[2]{%
  \glsdoifexists{#1}%
  {%
    \glsfieldfetch{#1}{#2}{\@gls@value}%
    \expandafter\glscapitalisewords\expandafter{\@gls@value}%
  }%
}
\newcommand*\glsentrytype}[1]{\@gls@entry@field{#1}{type}}
\newcommand*\glsentrysort}[1]{%
  \@gls@entry@field{#1}{sort}%
}
\newcommand*\glsentryparent}[1]{%
  \@gls@entry@field{#1}{parent}%
}
\newcommand*\glsentryuseri}[1]{%
  \@gls@entry@field{#1}{useri}%
}
\newcommand*\Glsentryuseri}[1]{%
  \@Glsentryfield{#1}{useri}%
}
\glsmfuaddmap{\glsentryuseri}{\Glsentryuseri}
\newcommand*\glsentryuserii}[1]{%
  \@gls@entry@field{#1}{userii}%
}
\newcommand*\Glsentryuserii}[1]{%
  \@Glsentryfield{#1}{userii}%
}
\glsmfuaddmap{\glsentryuserii}{\Glsentryuserii}
\newcommand*\glsentryuseriii}[1]{%
  \@gls@entry@field{#1}{useriii}%
}
\newcommand*\Glsentryuseriii}[1]{%
  \@Glsentryfield{#1}{useriii}%
}
\glsmfuaddmap{\glsentryuseriii}{\Glsentryuseriii}
\newcommand*\glsentryuseriv}[1]{%
  \@gls@entry@field{#1}{useriv}%
}
\newcommand*\Glsentryuseriv}[1]{%
  \@Glsentryfield{#1}{useriv}%
}
\glsmfuaddmap{\glsentryuseriv}{\Glsentryuseriv}
\newcommand*\glsentryuserv}[1]{%
  \@gls@entry@field{#1}{userv}%
}
}

```

```

\newcommand*{\Glsentryuser}[1]{%
  \@Glsentryfield{#1}{user}%
}
\glsmfuaddmap{\glsentryuser}{\Glsentryuser}
\newcommand*{\glsentryuseri}[1]{%
  \@gls@entry@field{#1}{useri}%
}
\newcommand*{\Glsentryuseri}[1]{%
  \@Glsentryfield{#1}{useri}%
}
\glsmfuaddmap{\glsentryuseri}{\Glsentryuseri}
\newcommand*{\glsentryshort}[1]{\@gls@entry@field{#1}{short}}
\newcommand*{\Glsentryshort}[1]{%
  \@Glsentryfield{#1}{short}%
}
\glsmfuaddmap{\glsentryshort}{\Glsentryshort}
\newcommand*{\glsentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}
\newcommand*{\Glsentryshortpl}[1]{%
  \@Glsentryfield{#1}{shortpl}%
}
\glsmfuaddmap{\glsentryshortpl}{\Glsentryshortpl}
\newcommand*{\glsentrylong}[1]{\@gls@entry@field{#1}{long}}
\newcommand*{\Glsentrylong}[1]{%
  \@Glsentryfield{#1}{long}%
}
\glsmfuaddmap{\glsentrylong}{\Glsentrylong}
\newcommand*{\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}
\newcommand*{\Glsentrylongpl}[1]{%
  \@Glsentryfield{#1}{longpl}%
}
\glsmfuaddmap{\glsentrylongpl}{\Glsentrylongpl}
\newcommand*{\glsentryfull}[1]{%
  \glsentrylong{#1} (\glsentryshort{#1})%
}
\newcommand*{\Glsentryfull}[1]{%
  \Glsentrylong{#1} (\glsentryshort{#1})%
}
\glsmfuaddmap{\glsentryfull}{\Glsentryfull}
\newcommand*{\glsentryfullpl}[1]{%
  \acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newcommand*{\Glsentryfullpl}[1]{%
  \Glsentrylongpl{#1} (\glsentryshortpl{#1})%
}
\glsmfuaddmap{\glsentryfullpl}{\Glsentryfullpl}
\newcommand*{\glsentrynumberlist}[1]{%
  \glsdoifexists{#1}%
  {%
    \@gls@entry@field{#1}{numberlist}%
  }%
}

```

```

}
\@ifpackageloaded{hyperref} {%
  \newcommand*\glsdisplaynumberlist}[1]{%
    \GlossariesWarning
    {%
      \string\glsdisplaynumberlist\space
      doesn't work with hyperref.^^JUsing
      \string\glsentrynumberlist\space instead%
    }%
    \glsentrynumberlist{#1}%
  }%
}%
{%
\newcommand*\glsdisplaynumberlist}[1]{%
  \glsdoifexists{#1}%
  {%
    \bgroup
      \protected@edef\@glo@label{\glsdetoklabel{#1}}%
      \let\@org@glsnumberformat\glsnumberformat
      \def\glsnumberformat##1{##1}%
      \protected@edef\the@numberlist{%
        \csname glo@\@glo@label @numberlist\endcsname}%
      \def\@gls@numlist@sep{}%
      \def\@gls@numlist@nextsep{}%
      \def\@gls@numlist@lastsep{}%
      \def\@gls@thislist{}%
      \def\@gls@donext@def{}%
      \renewcommand\do[1]{%
        \protected@edef\@gls@thislist{%
          \@gls@thislist
          \noexpand\@gls@numlist@sep
          ##1%
        }%
        \let\@gls@numlist@sep\@gls@numlist@nextsep
        \def\@gls@numlist@nextsep{\glsnumlistsep}%
        \@gls@donext@def
        \def\@gls@donext@def{%
          \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
        }%
      }%
      \expandafter \glsnumlistparser \expandafter{\the@numberlist}%
      \let\@gls@numlist@sep\@gls@numlist@lastsep
      \@gls@thislist
    \egroup
  }%
}
}
\newcommand*\glsnumlistsep}{, }
\newcommand*\glsnumlistlastsep}{ \& }
\newcommand*\gls hyperlink}[2][\glsentrytext{\@glo@label}]{%

```

```

\def\@glo@label{#2}%
\@glslink{\glo@linkprefix\glsdetoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*\glsadd[2][]{%
  \@gls@adjustmode
  \glsdoifexists{#2}%
  {%
    \def\@glsnumberformat{glsnumberformat}%
    \protected@edef\@gls@counter{\csname glo@\glsdetoklabel{#2}@counter\endcsname}%
    \setkeys{glossadd}{#1}%
    \@gls@saveentrycounter
    \@gls@setsort{#2}%
    \@do@wrglossary{#2}%
  }%
}
\glsmfuexcl{\glsadd}
\newcommand*\@gls@adjustmode{}
\AtBeginDocument{\renewcommand*\@gls@adjustmode{\ifvmode\mbox{}\fi}}
\newrobustcmd*\glsaddall[1][]{%
  \protected@edef\@glo@type{\@glo@types}%
  \setkeys{glossadd}{#1}%
  \forallglsentries[\@glo@type]{\@glo@entry}{%
    \glsadd[#1]{\@glo@entry}%
  }%
}
\newrobustcmd*\glsaddallunused[1][\@glo@types]{%
  \forallglsentries[#1]{\@glo@entry}%
  {%
    \ifglsused{\@glo@entry}{\glsadd[format=glsignore]{\@glo@entry}}%
  }%
}
\newcommand*\glsignore[1]{}
\edef\glsopenbrace{\expandafter@gobble\string\{ }
\edef\glsclosebrace{\expandafter@gobble\string\} }
\edef\glsbackslash{\expandafter@gobble\string\ }
\edef\glsquote#1{\string"#1\string"}
\edef\glspercentchar{\expandafter@gobble\string\% }
\edef\glsstildechar{\string~ }
\ifglsxindy
  \newcommand*\@glsfirstletter}{A}
\fi
\newcommand*\GlsSetXdyFirstLetterAfterDigits{%
  \@ifstar\s@GlsSetXdyFirstLetterAfterDigits@GlsSetXdyFirstLetterAfterDigits}
\ifglsxindy
  \newcommand*\@GlsSetXdyFirstLetterAfterDigits[1]{%
    \renewcommand*\@glsfirstletter}{#1}}
  \newcommand*\s@GlsSetXdyFirstLetterAfterDigits[1]{%
    \renewcommand*\@glsfirstletter}{#1}}

```

```

    \@onelevel@sanitize\@glsfirstletter
  }
\else
  \newcommand*\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \glsnoxywarning\GlsSetXdyFirstLetterAfterDigits}
  \newcommand*\s@GlsSetXdyFirstLetterAfterDigits}{%
    \@GlsSetXdyFirstLetterAfterDigits
  }
\fi
\ifglsxindy
  \newcommand*\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
\fi
\newcommand*\@GlsSetXdyNumberGroupOrder}[1]{%
  \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
}
\ifglsxindy
  \newcommand*\@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*\@xdynumbergrouporder}{#1}%
  }
  \newcommand*\s@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*\@xdynumbergrouporder}{#1}%
    \@onelevel@sanitize\@xdynumbergrouporder
  }
\else
  \newcommand*\@GlsSetXdyNumberGroupOrder}[1]{%
    \glsnoxywarning\GlsSetXdyNumberGroupOrder}
  \newcommand*\s@GlsSetXdyNumberGroupOrder}{%
    \@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*\@glsminrange}{2}
\ifglsxindy
  \newcommand*\@GlsSetXdyMinRangeLength}[1]{%
    \renewcommand*\@glsminrange}{#1}}
\else
  \newcommand*\@GlsSetXdyMinRangeLength}[1]{%
    \glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglsxindy
  \def\writeist{%
    \ifundef{\glswrite}{\newwrite\glswrite}{}%
    \@gls@addpredefinedattributes
    \openout\glswrite=\istfilename
    \write\glswrite{;; xindy style file created by the glossaries
      package}%
    \write\glswrite{;; for document '\jobname' on
      \the\year-\the\month-\the\day}%
    \write\glswrite{^^J; required styles^^J}
    \@for\@xdystyle:=\@xdyrequiredstyles\do{%
      \ifx\@xdystyle\@empty

```

```

\else
  \protected@write\glswrite{}\{(require
    \string"\xdystyle.xdy\string")\}%
\fi
}%
\write\glswrite{^^J%
  ; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%
\@for\@gls@classI:=\@gls@xdy@locationlist\do{%
  \protected@write\glswrite{}\{(define-location-class
    \string"\@gls@classI\string"^^J\space\space\space
    (
      :sep "{{"
      \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
      :sep "}"
    )
    ^^J\space\space\space
    :min-range-length \@glsminrange^^J%
  )
}%
{%
  \@for\@gls@classII:=\@gls@xdy@locationlist\do{%
    \protected@write\glswrite{}\{(define-location-class
      \string"\@gls@classII-\@gls@classI\string"
      ^^J\space\space\space
      (
        :sep "{"
        \csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
        :sep "}"
        \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
        :sep "}"
      )
      ^^J\space\space\space
      :min-range-length \@glsminrange^^J%
    )
  }%
}%
}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\glsseeformat\string"

```

```

        :close \string"{}\string"}}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
  :open \string"\string
  \glossarysection[\string\glossarytoctitle]{\string
  \glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
  {%
    \@for\@this@attr:=\@xdyattributelist\do{%
      \protected@write\glswrite{ }\string\providecommand*%
      \expandafter\string
      \csname glsX\@this@ctr X\@this@attr\endcsname[2]%
      {%
        \string\setentrycounter
        [\expandafter\@gobble\string\#1]{\@this@ctr}%
        \expandafter\string
        \csname\@this@attr\endcsname
        {\expandafter\@gobble\string\#2}%
      }%
    }%
  }%
}%
\write\glswrite{%
  \string\begin
  {theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
  \space\space:close \string"\glspercentchar\glstildechar n\string
  \end{theglossary}\string\glossarypostamble
  \glstildechar n\string" ^^J\space\space\space
  :tree}}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \glstildechar n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list
  :sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
  :sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF

```

```

\ifx\gls@suffiF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffiF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffiFF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffiFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
  :open-head \string"\string\glsgroupheading
  \glsopenbrace\string"^^J\space\space\space
  :close-head \string"\glsclosebrace\string")}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}%
\write\glswrite{\@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}%
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax
\ifundef{\glswrite}{\newwrite\glswrite}{}\relax
\openout\glswrite=\istfilename
\write\glswrite{\glspercentchar\space makeindex style file
  created by the glossaries package}
\write\glswrite{\glspercentchar\space for document
  '\jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\@gls@actualchar'}
\write\glswrite{encap '\@gls@encapchar'}
\write\glswrite{level '\@gls@levelchar'}
\write\glswrite{quote '\@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
  \glossarytoctitle]{\string\glossarytitle}\string
  \glossarypreamble\string\n\string\begin{theglossary}\string

```



```

    \glossaryheader\string\n\string"}
\write\glswrite{postamble \string"\string%\string\n\string
    \end{theglossary}\string\glossarypostamble\string\n
    \string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
    \string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}
\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
    \string"\string\relax \string\glsresetentrylist\string\n
    \string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
    \string"\string\relax \string\glsresetentrylist\string\n
    \string"}
\write\glswrite{delim_0 \string"\string\{\string
    \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string
    \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_2 \string"\string\{\string
    \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_t \string"\string\}\string}\string"}
\write\glswrite{delim_n \string"\string\delimN \string"}
\write\glswrite{delim_r \string"\string\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
    \string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
    \string"\string}\string\relax
    \string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"glscpositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF@empty
\else
    \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF@empty
\else
    \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}

```

```

\fi
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremake\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{
\ifglxsindy
\newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\newcommand*\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}%
\ifpackageloaded{tracklang}%
{%
\IfTrackedLanguage{german}%
{%
\def\@gls@extramakeindexopts{-g}%
}%
}%
}%
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkquote####1#1####2#1####3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={####1}%
\noexpand\ifx\noexpand\null####2\noexpand\null
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####2#1#1\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####2#1####3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%

```

```

\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
  \noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
    \noexpand\def\noexpand\@gls@checkedmkidx{%
      \noexpand\expandafter\noexpand\@gls@checkquote####1\noexpand\@nil
      #1#1\noexpand\null
      \noexpand\expandafter\noexpand\@gls@updatechecked
      \noexpand\@gls@checkedmkidx{####1}%
      \noexpand\def\noexpand\@gls@checkedmkidx{%
        \noexpand\expandafter\noexpand\@gls@checkesquote####1\noexpand\@nil
        \expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
        \noexpand\null
        \noexpand\expandafter\noexpand\@gls@updatechecked
        \noexpand\@gls@checkedmkidx{####1}%
        \noexpand\def\noexpand\@gls@checkedmkidx{%
          \noexpand\expandafter\noexpand\@gls@checkesactual####1\noexpand\@nil
          \noexpand\?\noexpand\?\noexpand\null
          \noexpand\expandafter\noexpand\@gls@updatechecked
          \noexpand\@gls@checkedmkidx{####1}%
          \noexpand\def\noexpand\@gls@checkedmkidx{%
            \noexpand\expandafter\noexpand\@gls@checkactual####1\noexpand\@nil
            \noexpand?\noexpand?\noexpand\null
            \noexpand\expandafter\noexpand\@gls@updatechecked
            \noexpand\@gls@checkedmkidx{####1}%
            \noexpand\def\noexpand\@gls@checkedmkidx{%
              \noexpand\expandafter\noexpand\@gls@checkbar####1\noexpand\@nil
              \noexpand|\noexpand|\noexpand\null
              \noexpand\expandafter\noexpand\@gls@updatechecked
              \noexpand\@gls@checkedmkidx{####1}%
              \noexpand\def\noexpand\@gls@checkedmkidx{%
                \noexpand\expandafter\noexpand\@gls@checkeschar####1\noexpand\@nil
                \noexpand\|\noexpand\|\noexpand\null
                \noexpand\expandafter\noexpand\@gls@updatechecked
                \noexpand\@gls@checkedmkidx{####1}%
                \noexpand\def\noexpand\@gls@checkedmkidx{%
                  \noexpand\expandafter\noexpand\@gls@checklevel####1\noexpand\@nil
                  \noexpand!\noexpand!\noexpand\null
                  \noexpand\expandafter\noexpand\@gls@updatechecked
                  \noexpand\@gls@checkedmkidx{####1}%
                  \noexpand\def\noexpand\@gls@checkedmkidx{%
                    \noexpand\expandafter\noexpand\@gls@checkescllevel####1\noexpand\@nil
                    \noexpand\!\noexpand!\noexpand\null
                    \noexpand\expandafter\noexpand\@gls@updatechecked
                    \noexpand\@gls@checkedmkidx{####1}%
                    \noexpand\def\noexpand\@gls@checkedmkidx{%
                      \noexpand\expandafter\noexpand\@gls@checkesc####1\noexpand\@nil
                      \@backslashchar\@backslashchar\noexpand\null
                      \noexpand\expandafter\noexpand\@gls@updatechecked
                      \noexpand\@gls@checkedmkidx{####1}%
                    }%
                  }%
                }%
              }%
            }%
          }%
        }%
      }%
    }%
  }%
}

```

```

}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
  \noexpand\def\noexpand\@gls@checkescquote####1%
    \expandonce{\csname#1\endcsname}####2\expandonce{\csname#1\endcsname}%
    ####3\noexpand\null{%
      \noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
      \noexpand\toks@={####1}%
      \noexpand\ifx\noexpand\null####2\noexpand\null
        \noexpand\ifx\noexpand\null####3\noexpand\null
          \noexpand\edef\noexpand\@gls@checkedmkidx{%
            \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
          \noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
          \noexpand\else
            \noexpand\edef\noexpand\@gls@checkedmkidx{%
              \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
              \noexpand\@gls@quotechar\noexpand\string\expandonce{%
                \csname#1\endcsname}\noexpand\@gls@quotechar
                \noexpand\@gls@quotechar\noexpand\string\expandonce{%
                  \csname#1\endcsname}\noexpand\@gls@quotechar}%
              \noexpand\def\noexpand\@gls@checkescquote{%
                \noexpand\@gls@checkescquote####3\noexpand\null}%
            \noexpand\fi
          \noexpand\else
            \noexpand\edef\noexpand\@gls@checkedmkidx{%
              \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
              \noexpand\@gls@quotechar\noexpand\string
                \expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
            \noexpand\ifx\noexpand\null####3\noexpand\null
              \noexpand\def\noexpand\@gls@checkescquote{%
                \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
                \expandonce{\csname#1\endcsname}\noexpand\null}%
            \noexpand\else
              \noexpand\def\noexpand\@gls@checkescquote{%
                \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
                ####3\noexpand\null}%
            \noexpand\fi
          \noexpand\fi
        \noexpand\@gls@checkescquote
      }%
    }%
  }%
\@gls@docheckquotedef
}
\newcommand*\@gls@nosetquote}[1]{\PackageError{glossaries}%
  {\string\GlsSetQuote\space not permitted here}%
  {\Move \string\GlsSetQuote\space earlier in the preamble, as
  soon as possible after glossaries.sty has been loaded}}
\fi
\newcommand*\@gls@extramakeindexopts}[1]{
\newcommand{\noist}{%

```

```

\@gls@addpredefinedattributes
\let\writeist\relax
}
\newcommand{\@gls@noist@toolate}{%
\GlossariesWarning{too late to use \string\noist. The style file
has already been written}%
}
\newcommand*{\@makeglossary}[1]{%
\ifglossaryexists{#1}%
{%
\ifglssavewrites
\expandafter\newtoks\csname glo@#1@filetok\endcsname
\else
\expandafter\newwrite\csname glo@#1@file\endcsname
\expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
\fi
\@gls@renewglossary
\writeist
}%
{%
\PackageError{glossaries}%
{Glossary type ‘#1’ not defined}%
{New glossaries must be defined before using \string\makeglossaries}%
}%
}
\newcommand*{\@glsopenfile}[2]{%
\immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
\PackageInfo{glossaries}{Writing glossary file
\jobname.\csname @glotype@#2@out\endcsname}%
}
\newcommand*{\@closegls}[1]{%
\closeout\csname glo@#1@file\endcsname
}
\ifglsexindy
\newcommand*{\@gls@automake}[1]{%
\ifglossaryexists{#1}
{%
\@closegls{#1}%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-M ord/letorder }}%
{\let\@gls@order\@empty}%
\gls@xdy@set@lang{#1}%
\ifdefvoid\gls@codepage
{\let\@gls@codepage\@empty}%
{\def\@gls@codepage{-C \gls@codepage\space}}%
\edef\@gls@dothiswrite{\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \gls@istfilebase\space

```

```

        \@gls@codepage
        -t \jobname.\csuse{@glotype@#1@log}
        -o \jobname.\csuse{@glotype@#1@in}
        \jobname.\csuse{@glotype@#1@out}}%
    }%
    \@gls@dothiswrite
}
{
    \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}
}
\else
\newcommand*{\@gls@automake}[1]{%
    \ifglossaryexists{#1}
    {%
        \@closegls{#1}%
        \ifdefstring{\glsorder}{letter}%
        {\def\@gls@order{-1 }}%
        {\let\@gls@order\@empty}%
        \edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
            -s \listfilename\space
            -t \jobname.\csuse{@glotype@#1@log}
            -o \jobname.\csuse{@glotype@#1@in}
            \jobname.\csuse{@glotype@#1@out}}%
        }%
        \@gls@dothiswrite
    }%
    {
        \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
    }%
}
\fi
\ifglsxindy
\newcommand*{\@gls@automake@immediate}[1]{%
    \ifglossaryexists{#1}
    {%
        \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
        {%
            \ifdefstring{\glsorder}{letter}%
            {\def\@gls@order{-M ord/letorder }}%
            {\let\@gls@order\@empty}%
            \gls@xdy@set@lang{#1}%
            \ifdefvoid\gls@codepage
            {\let\@gls@codepage\@empty}%
            {\def\@gls@codepage{-C \gls@codepage\space}}%
            \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
                -I xindy
                \@gls@order
                -L \@gls@langmod\space
                -M \gls@istfilebase\space

```

```

        \@gls@codepage
        -t \jobname.\csuse{@glotype@#1@log}
        -o \jobname.\csuse{@glotype@#1@in}
        \jobname.\csuse{@glotype@#1@out}}%
    }%
    \@gls@dothiswrite
}
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{
    \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
    \ifglossaryexists{#1}
    {%
        \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
        {%
            \ifdefstring{\glsorder}{letter}%
            {\def\@gls@order{-l }}%
            {\let\@gls@order\@empty}%
            \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order
            -s \istfilename\space
            -t \jobname.\csuse{@glotype@#1@log}
            -o \jobname.\csuse{@glotype@#1@in}
            \jobname.\csuse{@glotype@#1@out}}%
            }%
            \@gls@dothiswrite
        }%
        {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
        doesn't exist. Rerun may be required}}%
    }%
    {
        \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
    }%
}
}
\fi
\newcommand*{\@warn@nomakeglossaries}{}
\newcommand*{\@warn@nomakeglossaries}{\@warn@nomakeglossaries}
\newcommand{\@gls@automake@types}{\@glo@types}
\newcommand{\@gls@automake@immediate}{%
    \ifnum\gls@automake@nr=2\relax
    \protected@edef\gls@tmp{\@gls@automake@types}%
    \@for\@gls@type:=\gls@tmp\do{%
        \ifdefempty{\@gls@type}{%
            {\@gls@automake@immediate{\@gls@type}}%
        }%
    }%
}
\glsautomakefalse

```

```

\renewcommand*{\@gls@doautomake}{}%
\else
\ifdefempty\gls@automake@makegloss{%
{%
\IfFileExists{\jobname.aux}%
{\immediate\write18{\gls@automake@makegloss\space \jobname}}{%
\glsautomakefalse
\renewcommand*{\@gls@doautomake}{}%
}%
\fi
}
\newcommand*\makeglossaries{%
\ifglxindy
\def\glsindexingsetting{xindy}%
\else
\def\glsindexingsetting{makeindex}%
\fi
\@domakeglossaries
{%
\@gls@@automake@immediate
\ifundef\glswrite{\newwrite\glswrite}{%
\protected@write\@auxout{\string\providecommand\string\@glsorder[1]{}}
\protected@write\@auxout{\string\providecommand\string\@istfilename[1]{}}
\ifundef\@gls@extramakeindexopts
}{%
{%
\protected@write\@auxout{\string\providecommand
\string\@gls@extramakeindexopts[1]{}}
\protected@write\@auxout{\string\@gls@extramakeindexopts
{\@gls@extramakeindexopts}}%
}%
\protected@write\@auxout{\string\@istfilename{\istfilename}}%
\protected@write\@auxout{\string\@glsorder{\glsorder}}
\@for\@glo@type:=\@glo@types\do{%
\ifthenelse{\equal{\@glo@type}{}}{}}{%
\@makeglossary{\@glo@type}}%
}%
\renewcommand*\newglossary[4][{}]{%
\PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%
\let\@makeglossary\@gobble
\let\makeglossaries\relax
\@disable@onlypremakeg
\let\gls@checkseeallowed\relax
\let\warn@nomakeglossaries\relax
\let\warn@noprintglossary\@makeglossaries@warn@noprintglossary
\ifglssavenumberlist
\edef\@gls@dodolistparser{\noexpand\DeclareListParser

```



```

        {\noexpand\glsnumlistparser}{\delimN}}}%
    \@gls@dodedeflistparser
\fi
\let\makenoidxglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
    \let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
\ifglsautomake
    \renewcommand*{\@gls@doautomake}{%
        \protected@edef\gls@tmp{\@gls@automake@types}%
        \@for\@gls@type:=\gls@tmp\do{%
            \ifdefempty{\@gls@type}{%
                {\@gls@automake{\@gls@type}}%
            }%
        }%
    }%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}
\newcommand\@makeglossaries@warn@noprintglossary{%
    \ifdefstring{\@glo@types}{,}%
    {%
        \GlossariesWarningNoLine{No glossaries have been defined}%
    }%
    {%
        \GlossariesWarningNoLine{No \string\printglossary\space
            or \string\printglossaries\space
            found. ^^J(Remove \string\makeglossaries\space if you
            don't want any glossaries.) ^^JThis document will not
            have a glossary}%
    }%
}%
\AtEndDocument{%
    \warn@nomakeglossaries
    \warn@noprintglossary
}
\newcommand*{\makenoidxglossaries}{%
    \def\glsindexingsetting{noidx}%
    \@domakeglossaries
    {%
        \renewcommand{\@gls@noref@warn}[1]{%
            \GlossariesWarning{Empty glossary for
                \string\printnoidxglossary[type={##1}].
                Rerun may be required (or you may have forgotten to use
                commands like \string\gls)}%
        }%
        \let\@gls@checkmkidxchars\@gobble
        \glsesclocationsfalse
        \let\@do@wrglossary\gls@noidxglossary
    }%
}

```

```

\let@gls@getgrouptitle@gls@noidx@getgrouptitle
\let@gls@checkseeallowed@relax
\renewcommand{\@do@seeglossary}[2]{%
  \protected@edef@gls@label{\glsdetoklabel{##1}}%
  \protected@write@auxout{}{%
    \string@gls@reference
      {\csname glo@\@gls@label @type\endcsname}%
      {\@gls@label}%
      {%
        \string@glsseeformat##2}%
      }%
  }%
}%
\AtBeginDocument
{%
  \write@auxout{\string\providecommand\string@gls@reference[3]{}%
}%
\def\warn@noprintglossary{%
  \GlossariesWarningNoLine{No \string\printnoidxglossary\space
    or \string\printnoidxglossaries ^^J
    found. (Remove \string\makenoidxglossaries\space if you
    don't want any glossaries.)^^JThis document will not have a glossary}%
}%
\let\warn@nomakeglossaries@relax
\let\makeglossaries@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
  \let@glo@assign@sortkey@glo@assign@sortkey
  \def@glo@sorttype@glo@default@sorttype}%
}%
\renewcommand*\new@glossaryentry[2]{%
  \PackageError{glossaries}{Glossary entries must be
    defined in the preamble^^Jwhen you use
    \string\makenoidxglossaries}%
  {Either move your definitions to the preamble or use
    \string\makeglossaries}%
}%
\renewcommand*{\glsentrynumberlist}[1]{%
  \letcs{@gls@loclist}{glo@glsdetoklabel{##1}@loclist}%
  \ifdef@gls@loclist
  {%
    \glsnoidxloclist{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for '##1'. Either
        a rerun is required or you haven't referenced the entry}%
    }%
  }%
}%

```

```

\renewcommand*{\glsdisplaynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \def\@gls@noidxloclist@sep{%
      \def\@gls@noidxloclist@sep{%
        \def\@gls@noidxloclist@sep{%
          \glsnumlistsep
        }%
      }%
    }%
    \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
  }%
  \def\@gls@noidxloclist@finalsep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
  \@gls@noidxloclist@finalsep
  \@gls@noidxloclist@prev
}%
{%
  ??\glsdoifexists{##1}%
  {%
    \GlossariesWarning{Missing location list for ‘##1’. Either
      a rerun is required or you haven’t referenced the entry}%
  }%
}%
\renewcommand*{\glsnumberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo@glsdetoklabel{##1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
  \let\@gls@org@glsseeformat\glsseeformat
  \let\glsnoidxdisplayloc##2\relax
  \let\glsseeformat##3\relax
  \ifdef\@gls@loclist
  {%
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
  \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
  \let\glsseeformat\@gls@org@glsseeformat
}%
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
\@glo@check@sortallowed\makenoidxglossaries

```

```

\let\@gls@savenonumberlist\gls@noidx@nonumberlist
}%
}
\@onlypreamble{\makenoidxglossaries}
\newcommand*\glsnumberlistloop}[3]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
\newcommand*\glsnoidxnumberlistloophandler}[1]{%
  #1%
}
\newcommand*\@no@makeglossaries}{%
  \PackageError{glossaries}{You can't use both
  \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
  together.}%
}
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
    is required to make \string\printnoidxglossary[type=#1] work}%
}
\newcommand*\gls@noidxglossary}{%
  \protected@write\@auxout{\gls@wrglossdisablelocationcmds}{%
    \string\@gls@reference
    {\csname glo@\@gls@label @type\endcsname}%
    {\@gls@label}%
    {\string\glsnoidxdisplayloc
     {\@glo@counterprefix}%
     {\@gls@counter}%
     {\@glsnumberformat}%
     {\@gls@locref}}%
    }%
  }%
}
\AtEndDocument{%
  \gls@writefiles
}
\newcommand*\@gls@writefiles}{%
  \foralllglossaries{\@glo@type}{%
    \ifcsundef{glo@\@glo@type @filetok}%
    {%
      \def\gls@tmp{}%
    }%
    {%
      \edef\gls@tmp{\expandafter\the
        \csname glo@\@glo@type @filetok\endcsname}%
    }%
    \ifx\gls@tmp\@empty
      \ifx\@glo@type\glsdefaulttype
        \GlossariesWarningNoLine{Glossary '\@glo@type' has no

```

```

        entries.^^JRemember to use package option ‘nomain’ if
you
        don’t want to^^Juse the main glossary}%
    \else
        \GlossariesWarningNoLine{Glossary ‘\@glo@type’ has no
            entries}%
    \fi
    \else
        \@glsopenfile{\glswrite}{\@glo@type}%
        \immediate\write\glswrite{%
            \expandafter\the
                \csname glo@\@glo@type @filetok\endcsname}%
        \immediate\closeout\glswrite
    \fi
}%
}
\newcommand*{\gls@glossary}[1]{%
    \@gls@glossary{#1}%
}
\newcommand*{\@gls@glossary}[2]{%
    \if@gls@debug
        \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
    \fi
}
\newcommand{\@gls@renewglossary}{%
    \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
    \let\@gls@renewglossary\@empty
}
\newcommand*{\gls@wrglossary}[2]{%
    \ifglssavewrites
        \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
        \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
            \expandafter{\@gls@tmp^^J}%
    \else
        \ifcsdef{glo@#1@file}%
        {%
            \expandafter\protected@write\csname glo@#1@file\endcsname{%
                \gls@disablepagerefexpansion}{#2}%
        }%
        {%
            \ifignoredglossary{#1}{}%
            {%
                \GlossariesWarning{No file defined for glossary ‘#1’}%
            }%
        }%
    \fi
    \endgroup\@esphack
}
\newcommand*{\@do@wrglossary}[1]{%
    \gls@ifnotmeasuring{\glswriteentry{#1}{\@do@wrglossary{#1}}}%
}

```

```

}
\newcommand*\glswriteentry}[2]{%
  \ifglsindexonlyfirst
    \ifglsused{#1}{#2}%
  \else
    #2%
  \fi
}
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
\newcommand*\gls@disablepagerefexpansion{%
\glswrglossdisablelocationcmds
\ifglsesclocations
  \@for\@gls@this:=\gls@protected@pagefmts\do
  {%
    \expandafter\let\@gls@this\relax
  }%
\fi
}
\newcommand{\glswrglossdisablelocationcmds}{}
\newcommand*\gls@alphpage}{\@Alph\c@page}
\newcommand*\gls@Alphpage}{\@Alph\c@page}
\newcommand*\gls@numberpage}{\number\c@page}
\newcommand*\gls@arabicpage}{\@arabic\c@page}
\newcommand*\gls@romanpage}{\romannumeral\c@page}
\newcommand*\gls@Romanpage}{\@Roman\c@page}
\newcommand*\glsaddprotectedpagefmt}[1]{%
  \ifglsesclocations
    \protected@eappto\gls@protected@pagefmts{\expandonce{\csname gls#1page\endcsname}}%
    \csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
    \protected@eappto\@wrglossarynumberhook{%
      \noexpand\let\expandonce{\csname org@gls#1\endcsname}%
      \expandonce{\csname#1\endcsname}%
      \noexpand\def\expandonce{\csname#1\endcsname}{%
        \noexpand\@wrglossary@pageformat
        \expandonce{\csname gls#1page\endcsname}%
        \expandonce{\csname org@gls#1\endcsname}%
      }%
    }%
  \else
    \GlsWarnAddProtectedPageFmt
  \fi
}
\newcommand{\GlsWarnAddProtectedPageFmt}{%
  \GlossariesWarning{\string\glsaddprotectedpagefmt\space has no
  effect with esclocations=false}%
}
\newcommand*\@wrglossarynumberhook{}
\newcommand{\@wrglossary@pageformat}[3]{%
  \ifx#3\c@page #1\else #2#3\fi
}

```

```

}
\newcommand*{\@do@wrglossary}[1]{%
  \ifglsclocations
    \@do@esc@wrglossary{#1}%
  \else
    \@do@noesc@wrglossary{#1}%
  \fi
}
\newcommand{\glswrglossdisableanchorcmds}{\let\glstexorpdfstring\@secondoftwo}
\ifdef\texorpdfstring
  {\appto\glswrglossdisableanchorcmds{\let\texorpdfstring\@secondoftwo}}{}
\ifdef\pdfstringdefPreHook
  {\appto\glswrglossdisableanchorcmds{\pdfstringdefPreHook}}{}
\newcommand{\glswrglosslocationtarget}[1]{#1}
\newcommand{\glswrglosslocationtextfmt}[1]{#1}
\newcommand*{\@do@noesc@wrglossary}[1]{%
  \expandafter\def\expandafter\@glslocref\expandafter{\theglsentrycounter}%
  \expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
  \ifx\@glsHlocref\@glslocref
    \def\@glo@counterprefix{}%
  \else
    \xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
  \fi
  \protected@edef\@gls@label{\glsdetoklabel{#1}}%
  \@do@wrglossary
}
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodsfalse
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
  \begingroup
    \let\gls@orgthe\the
    \let\gls@orgnumber\number
    \let\gls@orgarabic\@arabic
    \let\gls@orgromannumeral\romannumeral
    \let\gls@orgalph\@alph
    \let\gls@orgAlph\@Alph
    \let\gls@orgRoman\@Roman
    \ifglswrallowprimitivemods
      \def\gls@the##1{%
        \ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
      \def\the{\expandafter\gls@the}%
      \def\gls@number##1{%
        \ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
      \def\number{\expandafter\gls@number}%
      \def\romannumeral##1{%
        \ifx##1\c@page \gls@romanpage \else \gls@orgromannumeral##1\fi}%
    \fi
    \def\@arabic##1{%
      \ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
    \def\@Roman##1{%

```

```

\ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
\ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
\ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\gls@locref{\theglsentrycounter}%
\endgroup
\@gls@checkmkidxchars\@gls@locref
\expandafter\ifx\theglsentrycounter\theglsentrycounter\relax
\def\@glo@counterprefix{%
\else
\protected@edef\@gls@Hlocref{\theglsentrycounter}%
\@gls@checkmkidxchars\@gls@Hlocref
\@xp@gls@getcounterprefix{\@gls@locref}{\@gls@Hlocref}%
\fi
\protected@edef\@gls@label{\gls@detoklabel{#1}}%
\@do@@wrglossary
}
\newcommand*{\@do@@wrglossary}{%
\ifglxindy
\expandafter\@glo@check@mkidxrangechar\@gls@numberformat\@nil
\def\@glo@range{%
\expandafter\if\@glo@prefix\relax
\def\@glo@range{:open-range}%
\else
\expandafter\if\@glo@prefix\relax
\def\@glo@range{:close-range}%
\fi
\fi
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
(indexentry :tkey (\csname glo@\@gls@label @index\endcsname)
:locref \string"\@glo@counterprefix}{\@gls@locref}\string" %
:attr \string"\@gls@counter\@glo@suffix\string"
\@glo@range
)
}%
\else
\@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@gls@numberformat}%
{\@glo@counterprefix}%
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
\string\glossaryentry{\csname glo@\@gls@label @index\endcsname
\@gls@encapchar\@glo@numfmt}{\@gls@locref}}%
\fi
}
\newcommand*\@xp@gls@getcounterprefix[2]{%
\bgroup
\gls@wrglossdisableanchorcmds
\protected@edef\@do@gls@getcounterprefix{%

```



```

        \noexpand\egroup
        \noexpand\@gls@getcounterprefix{#1}{#2}%
    }%
    \do@gls@getcounterprefix
}
\newcommand*\@gls@getcounterprefix[2]{%
    \def\@gls@thisloc{#1}\def\@gls@thisHloc{#2}%
    \ifx\@gls@thisloc\@gls@thisHloc
        \def\@glo@counterprefix{}%
    \else
        \def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
            \def\@glo@tmp{##2}%
            \ifx\@glo@tmp\@empty
                \def\@glo@counterprefix{}%
            \else
                \def\@glo@counterprefix{##1}%
            \fi
        }%
        \@gls@get@counterprefix#2.#1\end@getprefix
        \ifx\@glo@counterprefix\@empty
            \GlossariesWarning{Hyper target ‘#2’ can’t be formed by
                prefixing^^Jlocation ‘#1’. You need to modify the
                definition of \string\theH\@gls@counter^^Jotherwise you
                will get the warning: "‘name{\@gls@counter.#1}’ has been^^J
                referenced but does not exist"%
        \fi
    \fi
}
\newcommand{\@do@seeglossary}[2]{%
    \def\@gls@xref{#2}%
    \@onelevel@sanitize\@gls@xref
    \@gls@checkmkidxchars\@gls@xref
    \ifglsxindy
        \gls@glossary{\csname glo@#1@type\endcsname}{%
            (indexentry
                :key (\csname glo@#1@index\endcsname)
                :xref (\string"\@gls@xref\string")
                :attr \string"see\string"
            )
        }%
    \else
        \gls@glossary{\csname glo@#1@type\endcsname}{%
            \string\glossaryentry{\csname glo@#1@index\endcsname
                \@gls@encapchar glsseeformat\@gls@xref}{Z}}%
    \fi
}
\def\@gls@fixbraces#1#2#3\@nil{%
    \if#2[\relax
        \@gls@fixbraces#1#2#3\@end@fixbraces
    \else

```

```

    \def#1{#2#3}%
  \fi
}
\def\@gls@fixbraces#1[#2]#3\end@fixbraces{%
  \def#1{[#2]{#3}}%
}
\newrobustcmd*\glssee[3][\seename]{%
  \@do@seeglossary{#2}{#1}{#3}}
\newcommand*\@glssee[3][\seename]{%
  \glssee[#1]{#3}{#2}}
\newrobustcmd*\glsseeformat[3][\seename]{%
  \emph{#1} \glsseelist{#2}}
\newrobustcmd*\glsseelist[1]{%
  \let\@gls@dolast\relax
  \let\@gls@donext\relax
  \@for\@gls@thislabel:=#1\do{%
    \ifx\@xfor@nextelement\@nnil
      \@gls@dolast
    \else
      \@gls@donext
    \fi
    \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
    \let\@gls@dolast\glsseelastsep
    \let\@gls@donext\glsseesep
  }%
}
\newcommand*\glsseelastsep{\space\andname\space}
\newcommand*\glsseesep{, }
\newrobustcmd*\glsseeitem[1]{\glshyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*\glsseeitemformat[1]{%
  \ifglshashshort{#1}{\glsentrytext{#1}}{\glsentryname{#1}}%
}
\newcommand*\gls@save@numberlist[1]{%
  \ifglssavenumberlist
    \toks@{#1}%
    \protected@edef\@do@writeaux@info{%
      \noexpand\csgdef{glo@glscurrententrylabel @numberlist}{\the\toks@}%
    }%
    \@onelevel@sanitize\@do@writeaux@info
    \protected@write\@auxout{\@do@writeaux@info}%
  \fi
}
\newcommand*\warn@noprntglossary{}%
\ifcsundef{printglossary}{}%
{%
  \@gls@warnonglossdefined
  \undef\printglossary
}
\newcommand*\@printgloss@checkexists@noignored[2]{%
  \@ifglossaryexists{#1}%

```

```

{#2}%
{%
  \ifignoredglossary{#1}%
  {\GlossariesWarning{Glossary '#1' is an ignored glossary}}%
  {\GlossariesWarning{Glossary '#1' doesn't exist}}%
}%
}
\newcommand*{\@printgloss@checkexists@allowignored}[2]{%
  \s@ifglossaryexists{#1}%
  {#2}%
  {\GlossariesWarning{Glossary '#1' doesn't exist}}%
}
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\newcommand*{\printglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@glossary}%
}
\newcommand*{\printglossaries}{%
  \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}
\newcommand*{\printnoidxglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@noidx@glossary}%
}
\newcommand*{\printnoidxglossaries}{%
  \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
\newcommand*{\@printgloss@setsort}{}
\newcommand*{\@gls@preglossaryhook}{}
\newcommand{\@printglossary}[2]{%
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{\csname @glo@type\@glo@type @title\endcsname}%
  \def\glossarytoctitle{\glossarytitle}%
  \let\org@glossarytitle\glossarytitle
  \def\@glossarystyle{%
    \ifx\@glossary@default@style\relax
      \GlossariesWarning{No default glossary style provided \MessageBreak
        for the glossary '\@glo@type'. \MessageBreak
        Using fallback. \MessageBreak
        To fix this set the style with \MessageBreak
        \string\setglossarystyle\space or use the \MessageBreak
        style key=value option}%
    \fi
  }%
  \def\gls@dotoc@title{\glssettoc@title{\@glo@type}}%
  \let\@org@glossaryentrynumbers\glossaryentrynumbers
  \bgroup
  \@printgloss@setsort
  \setkeys{printgloss}{#1}%
  \@printgloss@checkexists{\@glo@type}%
}

```

```

{%
\ifx\glossarytitle\org@glossarytitle
\else
\expandafter\let\csname @glotype@\@glo@type @title\endcsname
\glossarytitle
\fi
\let\currentglossary\@glo@type
\let\org@glossaryentrynumbers\glossaryentrynumbers
\let\glsnonextpages\@glsnonextpages
\let\glsnextpages\@glsnextpages
\let\nopostdesc\@nopostdesc
\gls@dotoc@title
\@glossarystyle
\let\gls@org@glossaryentryfield\glossentry
\let\gls@org@glossarysubentryfield\subglossentry
\renewcommand{\glossentry}[1]{%
\protected@xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
\gls@org@glossaryentryfield{##1}%
}%
\renewcommand{\subglossentry}[2]{%
\protected@xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
\gls@org@glossarysubentryfield{##1}{##2}%
}%
\@gls@preglossaryhook
#2%
}%
\egroup
\global\let\glossaryentrynumbers\@org@glossaryentrynumbers
\global\let\warn@noprintglossary\relax
}
\newcommand{\@print@glossary}{%
\makeatletter
\@input@{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
\IfFileExists{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
{}%
{\null}%
\ifglsxindy
\gls@xdy@set@lang{\@glo@type}%
\edef\@do@auxoutstuff{%
\noexpand\AtEndDocument{%
\noexpand\immediate\noexpand\write\@auxout{%
\string\providecommand\string\@xdylanguage[2]{}}%
\noexpand\immediate\noexpand\write\@auxout{%
\string\@xdylanguage{\@glo@type}{\@gls@langmod}}%
}%
}%
\@do@auxoutstuff
\edef\@do@auxoutstuff{%
\noexpand\AtEndDocument{%
\noexpand\immediate\noexpand\write\@auxout{%

```

```

        \string\providecommand\string\@gls@codepage[2]{}%
        \noexpand\immediate\noexpand\write\@auxout{%
        \string\@gls@codepage{\@glo@type}{\gls@codepage}}%
    }%
}%
\do@auxoutstuff
\fi
\renewcommand*\@warn@nomakeglossaries{%
    \GlossariesWarningNoLine{\string\makeglossaries\space
    hasn't been used,^^Jthe glossaries will not be updated}%
}%
}
\newcommand*\@glo@sortentries[2]{%
    \glosortentrieswarning
    \def\@glo@sortinglist{}%
    \def\@glo@sortinghandler{#1}%
    \protected@edef\@glo@type{#2}%
    \forlistcsloop{\@glo@do@sortentries}{\glsref@#2}%
    \csdef{\glsref@#2}{%
    \@for\@this@label:=\@glo@sortinglist\do{%
        \xifinlistcs{\@this@label}{\glsref@#2}%
        {}%
        {%
            \listcsxadd{\glsref@#2}{\@this@label}%
        }%
        \ifcsdef{\@glo@sortingchildren@\@this@label}%
        {%
            \@glo@addchildren{#2}{\@this@label}%
        }%
        {}%
    }%
}%
}
\newcommand*\@glo@addchildren[2]{%
    \bgroup
    \letcs{\@glo@childlist}{\@glo@sortingchildren@#2}%
    \@for\@this@childlabel:=\@glo@childlist\do
    {%
        \xifinlistcs{\@this@childlabel}{\glsref@#1}%
        {}%
        {%
            \listcsxadd{\glsref@#1}{\@this@childlabel}%
        }%
        \ifcsdef{\@glo@sortingchildren@\@this@childlabel}%
        {%
            \@glo@addchildren{#1}{\@this@childlabel}%
        }%
        {}%
    }%
}%
\egroup

```

```

}
\newcommand*{\@glo@do@sortentries}[1]{%
  \ifglshasparent{#1}%
  {%
    \protected@edef\@glo@parent{\csuse{glo@\glsdetoklabel{#1}@parent}}%
    \ifcsundef{glo@sortingchildren@\@glo@parent}%
    {%
      \csdef{glo@sortingchildren@\@glo@parent}{}%
    }%
    {}%
    \expandafter\@glo@sortedinsert
    \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
    \xifinlistcs{\@glo@parent}{\glsref{\@glo@type}}%
    {%
    }%
    {%
      \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
    }%
  }%
  {}%
  \@glo@sortedinsert{\@glo@sortinglist}{#1}%
}%
}
\newcommand*{\@glo@sortedinsert}[2]{%
  \dtlinsertinto{#2}{#1}{\@glo@sortinghandler}%
}%
\newcommand*{\@glo@sorthandler@word}[2]{%
  \letcs@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce@gls@sort@B}%
    {\expandonce@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@letter}[2]{%
  \letcs@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce@gls@sort@B}%
    {\expandonce@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@case}[2]{%
  \letcs@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%

```

```

        \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}
\newcommand*{\@glo@sorthandler@nocase}[2]{%
    \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef\glo@do@compare{%
        \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}
\newcommand*{\@glo@sortmacro@word}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
        \@glo@sortentries{\@glo@sorthandler@word}{#1}%
    }%
    {%
        \PackageError{glossaries}{Conflicting sort options:^^J
            \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
            \string\printnoidxglossary[sort=word]}{-%
    }%
}
\newcommand*{\@glo@sortmacro@letter}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
        \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
    }%
    {%
        \PackageError{glossaries}{Conflicting sort options:^^J
            \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
            \string\printnoidxglossary[sort=letter]}{-%
    }%
}
\newcommand*{\@glo@sortmacro@standard}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
        \ifcsdef{\@glo@sorthandler@\glsorder}%
        {%
            \@glo@sortentries{\csuse{\@glo@sorthandler@\glsorder}}{#1}%
        }%
        {%
            \PackageError{glossaries}{Unknown sort handler ‘\glsorder’}{-%
        }%
    }%
    {%

```

```

        \PackageError{glossaries}{Conflicting sort options:^^J
        \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
        \string\printnoidxglossary[sort=standard]}{}}%
    }%
}
\newcommand*{\@glo@sortmacro@case}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
        \@glo@sortentries{\@glo@sorthandler@case}{#1}%
    }%
    {%
        \PackageError{glossaries}{Conflicting sort options:^^J
        \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
        \string\printnoidxglossary[sort=case]}{}}%
    }%
}
\newcommand*{\@glo@sortmacro@nocase}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
        \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
    }%
    {%
        \PackageError{glossaries}{Conflicting sort options:^^J
        \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
        \string\printnoidxglossary[sort=nocase]}{}}%
    }%
}
\newcommand*{\@glo@sortmacro@def}[1]{%
    \def\@glo@sortinglist{}%
    \forglseentries[#1]{\@gls@thislabel}%
    {%
        \xifinlistcs{\@gls@thislabel}{\@glsref@#1}%
        {%
            \listcsadd{\@glo@sortinglist}{\@gls@thislabel}%
        }%
    }%
    }%
    \cslet{\@glsref@#1}{\@glo@sortinglist}%
}
\newcommand*{\@glo@sortmacro@def@do}[1]{%
    \ifinlistcs{#1}{\@glsref@\@glo@type}%
    {}%
    {%
        \listcsadd{\@glsref@\@glo@type}{#1}%
    }%
    \ifcsdef{\@glo@sortingchildren@#1}%
    {%
        \@glo@addchildren{\@glo@type}{#1}%
    }%
}

```



```

    }%
}
\newcommand*{\@glo@sortmacro@use}[1]{%
\newcommand*{\@print@noidx@glossary}{%
\ifcsdef{\@glsref@\@glo@type}%
{%
\ifcsdef{\@glo@sortmacro@\@glo@sorttype}%
{%
\csuse{\@glo@sortmacro@\@glo@sorttype}{\@glo@type}%
}%
}%
\PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{%
}%
\glossarysection[\@glossarytoctitle]{\@glossarytitle}%
\glossarypreamble
\def\@gls@currentlettergroup{}%
\begin{theglossary}%
\glossaryheader
\glsresetentrylist
\forlistcsloop{\@gls@noidx@do}{\@glsref@\@glo@type}%
\end{theglossary}%
\glossarypostamble
}%
}%
\@gls@noref@warn{\@glo@type}%
}%
}
\def\@glo@grabfirst#1#2\@nil{%
\def\@gls@firsttok{#1}%
\ifdefempty\@gls@firsttok
{%
\def\@glo@thislettergrp{0}%
}%
}%
\@onelevel@sanitize\@gls@firsttok
\expandafter\@glo@grabfirst\@gls@firsttok{}{}\@nil
}%
}
\def\@glo@grabfirst#1#2\@nil{%
\ifdefempty\@glo@thislettergrp
{%
\def\@glo@thislettergrp{glssymbols}%
}%
}%
\count@=\ucode‘#1\relax
\ifnum\count@=0\relax
\def\@glo@thislettergrp{glssymbols}%
\else
\ifdefstring\@glo@sorttype{case}%
{%

```

```

        \count@=#1\relax
      }%
    {%
  }%
  \edef\@glo@thislettergrp{\the\count@}%
\fi
}%
}
\newcommand{\@gls@noidx@do}[1]{%
  \global\letcs{\@gls@loclist}{glo\glsdetoklabel{#1}@loclist}%
  \ifglshasparent{#1}%
  {%
    \gls@level=\csuse{glo\glsdetoklabel{#1}@level}\relax
    \ifdefvoid{\@gls@loclist}
    {%
      \subglossentry{\gls@level}{#1}{}%
    }%
    {%
      \subglossentry{\gls@level}{#1}%
      {%
        \glsnoidxprenumberlist{#1}%
        \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
      }%
    }%
  }%
}%
{%
  \letcs{\@gls@sort}{glo\glsdetoklabel{#1}@sort}%
  \expandafter\glo@grabfirst\@gls@sort{}{}\@nil
  \ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
  {}%
  {%
    \ifdefempty{\@gls@currentlettergroup}{}%
    {%
      \global\let\@glo@thislettergrp\@glo@thislettergrp
      \glsgroupskip
    }%
    \glsgroupheading{\@glo@thislettergrp}%
  }%
  \global\let\@gls@currentlettergroup\@glo@thislettergrp
  \ifdefvoid{\@gls@loclist}
  {%
    \glossentry{#1}{}%
  }%
  {%
    \glossentry{#1}%
    {%
      \glsnoidxprenumberlist{#1}%
      \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
    }%
  }%
}%

```

```

}%
}
\newcommand*\glsnoidxloclist[1]{%
  \def\@gls@noidxloclist@sep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxloclisthandler}{#1}%
}
\newcommand*\glsnoidxloclisthandler[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%
  \@gls@noidxloclist@sep
  #1%
  \def\@gls@noidxloclist@sep{\delimN}%
  \def\@gls@noidxloclist@prev{#1}%
  }%
}
\newcommand*\glsnoidxdisplayloclisthandler[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%
  \@gls@noidxloclist@sep
  \@gls@noidxloclist@prev
  \def\@gls@noidxloclist@prev{#1}%
  }%
}
\newcommand*\glsnoidxdisplayloc[4]{%
  \setentrycounter[#1]{#2}%
  \csuse{#3}{#4}%
}
\newcommand*\@gls@reference}[3]{%
  \glsdoifexistsorwarn{#2}%
  {%
  \ifcsundef{\@glsref@#1}{\csgdef{\@glsref@#1}{}}{%
  \ifinlistcs{#2}{\@glsref@#1}%
  }{%
  \listcsgadd{\@glsref@#1}{#2}}%
  \ifcsundef{glo@\glsdetoklabel{#2}@loclist}%
  {\csgdef{glo@\glsdetoklabel{#2}@loclist}{}}%
  }{%
  \listcsgadd{glo@\glsdetoklabel{#2}@loclist}{#3}%
  }%
}
\define@key{printgloss}{type}{\def\@glo@type{#1}}
\define@key{printgloss}{title}{%
\def\glossarytitle{#1}%
\let\gls@dotocitle\relax
}

```

```

\define@key{printgloss}{toctitle}{%
\def\glossarytoctitle{#1}%
\let\gls@dottoctitle\relax
}
\define@key{printgloss}{style}{%
\ifcsundef{@glsstyle@#1}%
{%
\PackageError{glossaries}%
{Glossary style ‘#1’ undefined}{}%
}%
{%
\def\@glossarystyle{\setglossarystyle{#1}}%
}%
}
\define@choicekey{printgloss}{numberedsection}%
[\gls@numberedsection@val\gls@numberedsection@nr]%
{false,nolabel,autolabel,nameref}[nolabel]%
{%
\ifcase\gls@numberedsection@nr\relax
\renewcommand*\@@glossarysecstar{*}%
\renewcommand*\@@glossaryseclabel{}%
\or
\renewcommand*\@@glossarysecstar{}%
\renewcommand*\@@glossaryseclabel{}%
\or
\renewcommand*\@@glossarysecstar{}%
\renewcommand*\@@glossaryseclabel{\label{\glsautoprefix\@glo@type}}%
\or
\renewcommand*\@@glossarysecstar{*}%
\renewcommand*\@@glossaryseclabel{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}}%
\fi
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
\csuse{glsnogroupskip#1}%
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
\csuse{glsnopostdot#1}%
}
\newcommand*\GlsEntryCounterLabelPrefix{glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
\csuse{glsentrycounter#1}%
\@gls@define@glossaryentrycounter
}
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
\csuse{glsesubentrycounter#1}%
\@gls@define@glossarysubentrycounter
}
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%

```

```

\ifglsonumberlist
  \def\glossaryentrynumbers##1{%
\else
  \def\glossaryentrynumbers##1{##1}%
\fi}
\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
\newcommand*\@glo@no@assign@sortkey}[1]{%
  \PackageError{glossaries}{‘sort’ key not permitted with
  \string\printglossary}%
  {The ‘sort’ key may only be used with \string\printnoidxglossary}%
}
\newcommand*\@glo@assign@sortkey}[1]{%
  \def\@glo@sorttype{#1}%
}
\newcommand*\@glsnonextpages}{%
  \gdef\glossaryentrynumbers##1{%
    \glsresetentrylist
  }%
}
\newcommand*\@glsnextpages}{%
  \gdef\glossaryentrynumbers##1{%
    ##1\glsresetentrylist}}
\newcommand*\glsresetentrylist}{%
  \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
\newcommand*\glsnonextpages}{}
\newcommand*\glsnextpages}{}
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
\newcommand*\glsresetsubentrycounter}{%
  \ifglssubentrycounter
    \setcounter{glossarysubentry}{0}%
  \fi
}
\newcommand*\glsresetentrycounter}{%
  \ifglsenentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*\glsstepentry}[1]{%
  \ifglsenentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
\newcommand*\glsstepsubentry}[1]{%
  \ifglssubentrycounter
    \protected@edef\currentglssubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
  \fi

```

```

}
\newcommand*\glsrefentry}[1]{%
\ifglentrycounter
\ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
\else
\ifglssubentrycounter
\ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
\else
\gls{#1}%
\fi
\fi
}
\newcommand*\glsentrycounterlabel}{%
\ifglentrycounter
\theglossaryentry.\space
\fi
}
\newcommand*\glssubentrycounterlabel}{%
\ifglssubentrycounter
\theglossarysubentry)\space
\fi
}
\newcommand*\glsentryitem}[1]{%
\ifglentrycounter
\glsstepentry{#1}\glsentrycounterlabel
\else
\glsresetsubentrycounter
\fi
}
\newcommand*\glssubentryitem}[1]{%
\ifglssubentrycounter
\glsstepsubentry{#1}\glssubentrycounterlabel
\fi
}
\ifcsundef{theglossary}%
{%
\newenvironment{theglossary}{}{}%
}%
{%
\@gls@warnontheglossdefined
\renewenvironment{theglossary}{}{}%
}
\newcommand*\glossaryheader{}
\newcommand*\glsstarget}[2]{\@glsstarget{\glo@linkprefix#1}{#2}}
\newcommand*\glossentry}[2]{\glossentryname{#1} #2}
\newcommand*\glossentryname}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
\expandafter\glsnamefont\expandafter{\glo@name}%

```

```

    }%
  }
  \newcommand*{\Glossentryname}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \glsnamefont{\Glsentryname{#1}}%
    }%
  }
  \glsmfuaddmap{\glossentryname}{\Glossentryname}
  \newcommand*{\glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \glsentrydesc{#1}%
    }%
  }
  \newcommand*{\Glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \Glsentrydesc{#1}%
    }%
  }
  \glsmfuaddmap{\glossentrydesc}{\Glossentrydesc}
  \newcommand*{\glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \glsentrysymbol{#1}%
    }%
  }
  \newcommand*{\Glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \Glsentrysymbol{#1}%
    }%
  }
  \glsmfuaddmap{\glossentrysymbol}{\Glossentrysymbol}
  \newcommand{\subglossentry}[3]{\glossentryname{#2} #3}
  \newcommand*{\glsgroupskip}{=}
  \newcommand*{\glsgroupheading}[1]{=}
  \newrobustcmd*{\glsgetgrouptitle}[1]{%
    \@gls@getgrouptitle{#1}{\@gls@grptitle}%
    \@gls@grptitle
  }
  \ExplSyntaxOn
  \newcommand*{\@gls@getgrouptitle}[2]{%
    \def#2{#1}%
    \regex_match:nnT { ~\A(.|glsnumbers|glssymbols)\Z } { #1 }
    { \ifcsundef{#1groupname}{-}{\letcs#2{#1groupname}} }
  }
  \ExplSyntaxOff
  \newcommand*{\@gls@noidx@getgrouptitle}[2]{%

```

```

\DTLifint{#1}%
{\edef#2{\char#1\relax}}%
{%
  \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
}%
}
\ExplSyntaxOn
\newcommand*{\glsgetgrouplabel}[1]{%
  \str_if_eq:eeTF {#1} { \glssymbolsgroupname }
  { \glssymbols }
  {
    \str_if_eq:eeTF {#1} { \glsnumbersgroupname }
    { \glsnumbers }
    { #1 }
  }
}
\ExplSyntaxOff
\newcommand*{\setentrycounter}[2][1]{%
  \def\@glo@counterprefix{#1}%
  \ifx\@glo@counterprefix\empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1.}%
  \fi
  \def\glsentrycounter{#2}%
}
\newcommand*{\setglossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \csname @glsstyle@#1\endcsname
  }%
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\newcommand{\newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
\newcommand{\renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%

```



```

    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
\newcommand*{\glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\glsnumber#1{#1}%
}%
{%
  \def\glsnumber#1{\xp@glsnumber{#1}}
}
\newrobustcmd{\xp@glsnumber}[1]{%
  \glsnumber#1\nohyperpage{ }\@nil
}
\def\@glsnumber#1\nohyperpage#2#3\@nil{%
  \ifx\#1\%
    \else
      \gls@delimR#1\delimR\delimR\%
    \fi
  \ifx\#2\%
    \else
      #2%
    \fi
  \ifx\#3\%
    \else
      \glsnumber#3\@nil
    \fi
}
\def\gls@delimR#1\delimR #2\delimR #3\%
\ifx\#2\%
  \gls@delimN{#1}%
\else
  \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
\fi
}
\def\gls@delimN#1{\gls@delimN#1\delimN \delimN\}
\def\gls@delimN#1\delimN #2\delimN#3\%
\ifx\#3\%
  \@gls@numberlink{#1}%
\else
  \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
\fi
}
\def\@gls@numberlink#1{%
  \begingroup
  \toks@={}%
  \@gls@removespaces#1 \@nil

```

```

\endgroup
}
\def\@gls@removespaces#1 #2\@nil{%
\toks@=\expandafter{\the\toks@#1}%
\ifx\#2\%
\edef\@gls@x{\the\toks@}%
\ifx\@gls@x\empty
\else
\bgroup
\glswrglossdisableanchorcmds
\protected@edef\@gls@tmp{\glsentrycounter\@glo@counterprefix
\glswrglosslocationtarget{\@gls@x}}%
\@onelevel@sanitize\@gls@tmp
\edef\@gls@x{\noexpand\egroup
\noexpand\hyperlink{\@gls@tmp}%
{noexpand\glswrglosslocationtextfmt{\the\toks@}}%
}%
\@gls@x
\fi
\else
\@gls@ReturnAfterFi{%
\@gls@removespaces#2\@nil
}%
\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
\newcommand*\hyperrm}[1]{\textrm{\glsnumber{#1}}}
\newcommand*\hypersf}[1]{\textsf{\glsnumber{#1}}}
\newcommand*\hypertt}[1]{\texttt{\glsnumber{#1}}}
\newcommand*\hyperbf}[1]{\textbf{\glsnumber{#1}}}
\newcommand*\hypermd}[1]{\textmd{\glsnumber{#1}}}
\newcommand*\hyperit}[1]{\textit{\glsnumber{#1}}}
\newcommand*\hypersl}[1]{\textsl{\glsnumber{#1}}}
\newcommand*\hyperup}[1]{\textup{\glsnumber{#1}}}
\newcommand*\hypersc}[1]{\textsc{\glsnumber{#1}}}
\newcommand*\hyperemph}[1]{\emph{\glsnumber{#1}}}
\newcommand{\oldacronym}[4][\gls@label]{%
\def\gls@label{#2}%
\newacronym[#4]{#1}{#2}{#3}%
\ifcsundef{xspace}%
{%
\expandafter\edef\csname#1\endcsname{%
\noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}%
}%
}%
}
\def\@gls@label{#2}%
\newacronym[#4]{#1}{#2}{#3}%
\ifcsundef{xspace}%
{%
\expandafter\edef\csname#1\endcsname{%
\noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
\noexpand\gls{#1}\noexpand\xspace}%
}%
}

```

```

}%
}
\newcommand{\newacronym}[4][\%
\gls@newacronym{#1}{#2}{#3}{#4}%
}
\newcommand{\gls@newacronym}[4]{%
\renewcommand{\gls@newacronym}[4]{%
\PackageError{glossaries}{incompatible acronym style:
rollback may be required}%
{something has gone wrong with \string\newacronym. Try
\string\usepackage{glossaries}[=v4.49]}%
}%
\setacronymstyle{long-short}%
\newacronym[#1]{#2}{#3}{#4}%
}
\newcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}
\newrobustcmd*{\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
\newcommand*{\glsshortkey}{short}
\newcommand*{\glsshortpluralkey}{shortplural}
\newcommand*{\glslongkey}{long}
\newcommand*{\glslongpluralkey}{longplural}
\newcommand{\acrlinkfullformat}[5]{%
\acrfullformat{#1}{#3}{#4}[#5]{#2}{#3}{#4}[]}%
}
\newcommand{\acrfullformat}[2]{#1\glspace{#2}}
\newrobustcmd*{\glsspace}{\space}
\newcommand*{\acrfullfmt}[3]{%
\acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newcommand*{\Acrfullfmt}[3]{%
\acrlinkfullformat{\@Acrlong}{\@Acrshort}{#1}{#2}{#3}%
}
\newcommand*{\ACRfullfmt}[3]{%
\acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
\newcommand*{\acrfullplfmt}[3]{%
\acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newcommand*{\Acrfullplfmt}[3]{%
\acrlinkfullformat{\@Acrlongpl}{\@Acrshortpl}{#1}{#2}{#3}%
}
\newcommand*{\ACRfullplfmt}[3]{%
\acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfull}{\@gls@hyp@opt\ns@acrfull}
\newcommand*{ns@acrfull}[2][\%
\new@ifnextchar[{\@acrfull{#1}{#2}}%
{\@acrfull{#1}{#2}[]}%
}
\def\@acrfull#1#2[#3]{%

```

```

\acrfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\Acrfull}{\@gls@hyp@opt\ns@Acrfull}
\glsmfuaddmap{\acrfull}{\Acrfull}
\newcommand*\ns@Acrfull[2][]{%
  \new@ifnextchar[{\@Acrfull{#1}{#2}}%
    {\@Acrfull{#1}{#2}[]}%
}
\def\@Acrfull#1#2[#3]{%
  \Acrfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfull}{\@gls@hyp@opt\ns@ACRfull}
\newcommand*\ns@ACRfull[2][]{%
  \new@ifnextchar[{\@ACRfull{#1}{#2}}%
    {\@ACRfull{#1}{#2}[]}%
}
\def\@ACRfull#1#2[#3]{%
  \ACRfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfullpl}{\@gls@hyp@opt\ns@acrfullpl}
\newcommand*\ns@acrfullpl[2][]{%
  \new@ifnextchar[{\@acrfullpl{#1}{#2}}%
    {\@acrfullpl{#1}{#2}[]}%
}
\def\@acrfullpl#1#2[#3]{%
  \acrfullplfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\Acrfullpl}{\@gls@hyp@opt\ns@Acrfullpl}
\glsmfuaddmap{\acrfullpl}{\Acrfullpl}
\newcommand*\ns@Acrfullpl[2][]{%
  \new@ifnextchar[{\@Acrfullpl{#1}{#2}}%
    {\@Acrfullpl{#1}{#2}[]}%
}
\def\@Acrfullpl#1#2[#3]{%
  \Acrfullplfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfullpl}{\@gls@hyp@opt\ns@ACRfullpl}
\newcommand*\ns@ACRfullpl[2][]{%
  \new@ifnextchar[{\@ACRfullpl{#1}{#2}}%
    {\@ACRfullpl{#1}{#2}[]}%
}
\def\@ACRfullpl#1#2[#3]{%
  \ACRfullplfmt{#1}{#2}{#3}%
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand*\acrnameformat[2]{\acronymfont{#1}}
\newtoks\glskeylisttok
\newtoks\glslabeltok
\newtoks\glsshorttok

```

```

\newtoks\glslongtok
\newcommand*{\newacronymhook}{}
\newcommand*{\SetGenericNewAcronym}{%
  \let\@Gls@entryname\@Gls@acentryname
  \renewcommand{\newacronym}[4][]{%
    \ifdefempty{\@glsacronymlists}%
      {%
        \def\@glo@type{\acronymtype}%
        \setkeys{glossentry}{##1}%
        \DeclareAcronymList{\@glo@type}%
      }%
    }%
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \glslongtok{##4}%
  \newacronymhook
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\expandonce{\acronymentry{##2}}},%
      sort={\acronymsort{\the\glsshorttok}{\the\glslongtok}},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\glspluralsuffix},%
      \GenericAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsuppercase{\genacrfullformat{##2}{##3}}}%
}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsuppercase{\genplacrfullformat{##2}{##3}}}%
}%
\renewcommand*{\glsentryfull}[1]{\genacrfullformat{##1}{}}%
\renewcommand*{\Glsentryfull}[1]{\Genacrfullformat{##1}{}}%

```

```

\renewcommand*{\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}%
\renewcommand*{\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}%
}
\newcommand*{\GenericAcronymFields}{description={\the\glslongtok}}
\newcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
\newcommand*{\acronymsort}[2]{#1}
\newcommand*{\setacronymstyle}[1]{%
\ifcsundef{@glsacr@dispstyle@#1}
{%
\PackageError{glossaries}{Undefined acronym style ‘#1’}{}%
}%
{%
\ifdefempty{\@glsacronymlists}%
{%
\DeclareAcronymList{\acronymtype}%
}%
{}%
\SetGenericNewAcronym
\GlsUseAcrStyleDefs{##1}%
\@for\@gls@type:=\@glsacronymlists\do{%
\defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{##1}}%
}%
}%
}
\newcommand*{\newacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@#1}%
{%
\PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
}%
{%
\csdef{@glsacr@dispstyle@#1}{#2}%
\csdef{@glsacr@styledefs@#1}{#3}%
}%
}
\newcommand*{\renewacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@#1}%
{%
\csdef{@glsacr@dispstyle@#1}{#2}%
\csdef{@glsacr@styledefs@#1}{#3}%
}%
{%
\PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
}%
}
\newcommand*{\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*{\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
{%
\ifglsashaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%

```

```

{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
\ifglsashaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
}

```

```

\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newcommand*{\glsacspace}[1]{%
\settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{##1}})}%
\ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
\ifglsahaslong{\glslabel}{\glsacspace}{\glsacspace}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshort{##1}}##2\space
(\glsentrylong{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshort{##1}}##2\space
(\glsentrylong{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
(\glsentrylongpl{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
(\glsentrylongpl{##1})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{%
\GlsUseAcrEntryDispStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{long-sm-short}%
{%
\GlsUseAcrEntryDispStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
}

```



```

\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{sc-short-long}%
{
\GlsUseAcrEntryDispStyle{short-long}%
}%
{
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{sm-short-long}%
{
\GlsUseAcrEntryDispStyle{short-long}%
}%
{
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{
\GlsUseAcrEntryDispStyle{long-short}%
}%
{
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sp-short-desc}%
{
\GlsUseAcrEntryDispStyle{long-sp-short}%
}%
{
\GlsUseAcrStyleDefs{long-sp-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\glsacspace{##1}(\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sc-short-desc}%
{
\GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{
\GlsUseAcrStyleDefs{long-sc-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
}

```

```

\renewcommand*\acronymentry}[1]{%
  \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext

```

```

{%
  \ifglshaslong{\glslabel}%
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \glsentrylongpl{\glslabel}\glsinsert
      }%
      {%
        \Glsentrylongpl{\glslabel}\glsinsert
      }%
      {%
        \glsupercase
        {\glsentrylongpl{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glscapscase
    {%
      \glsentrylong{\glslabel}\glsinsert
    }%
    {%
      \Glsentrylong{\glslabel}\glsinsert
    }%
    {%
      \glsupercase
      {\glsentrylong{\glslabel}\glsinsert}%
    }%
  }%
  }%
  {%
    \glsgenentryfmt
  }%
  }%
  {\glscustomtext\glsinsert}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\acrfullfmt}[3]{%
    \glslink[##1]{##2}{\glsentrylong{##2}##3\space
      (\acronymfont{\glsentryshort{##2}})}%
  }%
  \renewcommand*{\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{\Glsentrylong{##2}##3\space
      (\acronymfont{\glsentryshort{##2}})}%
  }%
  \renewcommand*{\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
      \glsupercase{\glsentrylong{##2}##3\space
        (\acronymfont{\glsentryshort{##2}})}%
    }%
  }%
  \renewcommand*{\acrfullplfmt}[3]{%

```

```

\glslink[##1]{##2}{\glsentrylongpl{##2}##3\space
(\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{\Glsentrylongpl{##2}##3\space
(\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%
\glsupercase{\glsentrylongpl{##2}##3\space
(\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\glsentryfull}[1]{%
\glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
\Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
\glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
\Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
\GlsUseAcrEntryDispStyle{dua}%
}%
{%
\GlsUseAcrStyleDefs{dua}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymentry}[1]{\glsentrylong{##1}}%
\renewcommand*{\acronymsort}[2]{##2}%
}%
\newacronymstyle{footnote}%
{%
\ifglsashaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshort{##1}}##2%
\protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\firstacronymfont{\Glsentryshort{##1}}##2%
\protect\footnote{\Glsentrylong{##1}}%
}

```

```

}%
\renewcommand*\genplacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\acrpluralsuffix{\glsacrpluralsuffix}%
\renewcommand*\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
  \glsupercase{\acronymfont{\glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}}}%
\renewcommand*\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
  \glsupercase{\acronymfont{\glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}}}%
\renewcommand*\glsentryfull}[1]{%
  \acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*\Glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*\glsentryfullpl}[1]{%
  \acronymfont{\glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
\renewcommand*\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
}
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymfont}[1]{\textsc{##1}}%

```

```

\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}%
\newacronymstyle{footnote-sm}%
{%
\GlsUseAcrEntryDispStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}%
\newacronymstyle{footnote-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{footnote-sc-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
\GlsUseAcrStyleDefs{footnote-sc}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{footnote-sm-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote-sm}%
}%
{%
\GlsUseAcrStyleDefs{footnote-sm}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newcommand*{\DefineAcronymSynonyms}{%
\let\acs\acrshort
\let\Acs\Acrshort
\let\acsp\acrshortpl
\let\Acsp\Acrshortpl

```

```

\let\acl\acrlong
\let\aclp\acrlongpl
\let\Acl\Acrlong
\let\Aclp\Acrlongpl
\let\acf\acrfull
\let\acfp\acrfullpl
\let\Acf\Acrfull
\let\Acfp\Acrfullpl
\let\ac\gls
\let\Ac\Gls
\let\acp\glspl
\let\Acp\Glspl
}
\ifglsacrshortcuts
  \DefineAcronymSynonyms
\fi
\newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
\newcommand*{\acrlinkfootnote}[3]{%
  \footnote{\glslink[#1]{#2}{#3}}%
}
\newcommand*{\acrnoflinkfootnote}[3]{%
  \footnote{#3}%
}
\newrobustcmd*{\glsdoparenifnotempty}[2]{%
  \protected@edef\gls@tmp{#1}%
  \ifdefempty\gls@tmp
  {}%
  {%
    \ifx\gls@tmp\@gls@default@value
    \else
    \space (#2{#1})%
    \fi
  }%
}
\newcommand*{\@gls@deprecated@acr@warning}[3]{%
  \GlossariesWarning{Acronym style option #1 deprecated.
  Use \string\setacronymstyle{#2} with just glossaries.sty or
  use \string\setabbreviationstyle[acronym]{#3} with
  glossaries-extra.sty}%
}
\RequirePackage{glossary-hypernav}[=v4.52]
\@gls@loadlist
\@gls@loadlong
\@gls@loadsuper
\@gls@loadtree
\ifx\@glossary@default@style\relax
\else

```

```

\setglossarystyle{\@glossary@default@style}
\fi
\newcommand*{\showgloparent}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
}
\newcommand*{\showglolevel}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
}
\newcommand*{\showglotext}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
}
\newcommand*{\showgloplural}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@plural\endcsname
}
\newcommand*{\showglofirst}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@first\endcsname
}
\newcommand*{\showglofirstpl}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@firstpl\endcsname
}
\newcommand*{\showglotype}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@type\endcsname
}
\newcommand*{\showglocounter}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@counter\endcsname
}
\newcommand*{\showglouseri}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@useri\endcsname
}
\newcommand*{\showglouserii}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@userii\endcsname
}
\newcommand*{\showglouseriii}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@useriii\endcsname
}
\newcommand*{\showglouseriv}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@useriv\endcsname
}
\newcommand*{\showglouserv}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@userv\endcsname
}
\newcommand*{\showglouservi}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@uservi\endcsname
}
\newcommand*{\showglongame}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@name\endcsname
}
\newcommand*{\showglodesc}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@desc\endcsname
}
}

```



```

\newcommand*{\showgloDESCplural}[1]{%
  \expandafter\show\csname glo@glSDetoklabel{#1}@DESCplural\endcsname
}
\newcommand*{\showgloSort}[1]{%
  \expandafter\show\csname glo@glSDetoklabel{#1}@Sort\endcsname
}
\newcommand*{\showgloSymbol}[1]{%
  \expandafter\show\csname glo@glSDetoklabel{#1}@Symbol\endcsname
}
\newcommand*{\showgloSymbolplural}[1]{%
  \expandafter\show\csname glo@glSDetoklabel{#1}@Symbolplural\endcsname
}
\newcommand*{\showgloShort}[1]{%
  \expandafter\show\csname glo@glSDetoklabel{#1}@Short\endcsname
}
\newcommand*{\showgloLong}[1]{%
  \expandafter\show\csname glo@glSDetoklabel{#1}@Long\endcsname
}
\newcommand*{\showgloIndex}[1]{%
  \expandafter\show\csname glo@glSDetoklabel{#1}@Index\endcsname
}
\newcommand*{\showgloFlag}[1]{%
  \expandafter\show\csname ifglo@glSDetoklabel{#1}@Flag\endcsname
}
\newcommand*{\showgloLoclist}[1]{%
  \expandafter\show\csname glo@glSDetoklabel{#1}@Loclist\endcsname
}
\newcommand*{\showgloField}[2]{%
  \csshow{glo@glSDetoklabel{#1}@#2}%
}
\newcommand*{\showacronymlists}{%
  \show\@glSACronymlists
}
\newcommand*{\showglossaries}{%
  \show\@glo@types
}
\newcommand*{\showglossaryin}[1]{%
  \expandafter\show\csname @glo@type@#1@in\endcsname
}
\newcommand*{\showglossaryout}[1]{%
  \expandafter\show\csname @glo@type@#1@out\endcsname
}
\newcommand*{\showglossarytitle}[1]{%
  \expandafter\show\csname @glo@type@#1@title\endcsname
}
\newcommand*{\showglossarycounter}[1]{%
  \expandafter\show\csname @glo@type@#1@counter\endcsname
}
\newcommand*{\showglossaryentries}[1]{%
  \expandafter\show\csname glo@list@#1\endcsname
}

```

```
}
```

8.36 Rollback v4.52 (glossary-hypernav-2022-11-03.sty)

Version 4.52 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossary-hypernav}[2022/11/03 v4.52 (NLCT)]
\newcommand*\glsnavhyperlink}[3][\currentglossary]{%
  \protected@edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%
  \glslink{\glsnavhyperlinkname{#1}{#2}}{#3}%
}
\newcommand*\glsnavhyperlinkname}[2]{glsn:#1@#2}
\newcommand*\glsnavhypertarget}{\protect\@gls@navhypertarget}
\newcommand*\@gls@navhypertarget}[3][\currentglossary]{%
  \glsnavhypertarget{#1}{#2}{#3}%
}
\newcommand*\@gls@navhypertarget}[3]{%
  \protected@write\@auxout{}{\string\@gls@hypergroup{#1}{#2}}%
  \gls@target{\glsnavhyperlinkname{#1}{#2}}{#3}%
  \expandafter\let
    \expandafter\@gls@list\csname @gls@hypergroup@list@#1\endcsname
  \@for\@gls@elem:=\@gls@list\do{%
    \ifthenelse{equal{\@gls@elem}{#2}}{\@endfortrue}{}%
  }
  \if@endfor
  \else
    \GlossariesWarningNoLine{Navigation panel
      for glossary type ‘#1’^^Jmissing group ‘#2’}%
    \gdef\gls@hypergroup@rerrun{%
      \GlossariesWarningNoLine{Navigation panel
        has changed. Rerun LaTeX}}%
  \fi
}
\let\gls@hypergroup@rerrun\relax
\AtEndDocument{\gls@hypergroup@rerrun}
\newcommand*\@gls@hypergroup}[2]{%
  \ifcsdef{\@gls@hypergroup@list@#1}%
  {\protected@csxappto{\@gls@hypergroup@list@#1}{, #2}}%
  {\protected@csxdef{\@gls@hypergroup@list@#1}{#2}}%
}
\newcommand*\glsnavigation}{%
  \def\@gls@between{}%
  \ifcsundef{\@gls@hypergroup@list@\currentglossary}%
  {%
    \def\@gls@list{}%
  }%
  {%
    \letcs\@gls@list{\@gls@hypergroup@list@\currentglossary}%
  }%
  \@for\@gls@tmp:=\@gls@list\do{%
    \@gls@between
```



```

\RequirePackage{amsgen}
\RequirePackage{etoolbox}
\newif\if@gls@docloaded
\@ifpackageloaded{doc}{%
{%
  \@gls@docloadedtrue
}%
}%
{%
  \@ifclassloaded{nlctdoc}{\@gls@docloadedtrue}{\@gls@docloadedfalse}%
}
\if@gls@docloaded
  \let\glsorg@theglossary\theglossary
  \let\glsorg@endtheglossary\endtheglossary
  \let\glsorg@PrintChanges\PrintChanges
  \renewcommand{\PrintChanges}{%
    \begingroup
      \let\theglossary\glsorg@theglossary
      \let\endtheglossary\glsorg@endtheglossary
      \glsorg@PrintChanges
    \endgroup
  }
\fi
\newif\if@gls@debug
\define@choicekey{glossaries.sty}{debug}{\@gls@debug@val\@gls@debug@nr}{%
{true,false,showtargets,showaccsupp}[true]{%
  \ifcase\@gls@debug@nr\relax
  % debug=true
  \@gls@debugtrue
  \renewcommand*\@GlossariesWarning}[1]{%
    \PackageWarning{glossaries}{##1}%
  }%
  \renewcommand*\@GlossariesWarningNoLine}[1]{%
    \PackageWarningNoLine{glossaries}{##1}%
  }%
  \let\@glsshowtarget\@gobble
  \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\or
  % debug=false
  \@gls@debugfalse
  \let\@glsshowtarget\@gobble
  \let\@glsshowaccsupp\@gobblethree
  \PackageInfo{glossaries}{debug mode OFF}%
\or
  % debug=showtargets
  \@gls@debugtrue
  \renewcommand*\@GlossariesWarning}[1]{%
    \PackageWarning{glossaries}{##1}%
  }%
  \renewcommand*\@GlossariesWarningNoLine}[1]{%
    \PackageWarningNoLine{glossaries}{##1}%
  }%
}

```

```

    }%
    \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
    \renewcommand{\@glsshowtarget}{\@@glsshowtarget}%
  \or
  % debug=showaccsupp
  \@gls@debugtrue
  \renewcommand*\GlossariesWarning}[1]{%
    \PackageWarning{glossaries}{##1}%
  }%
  \renewcommand*\GlossariesWarningNoLine}[1]{%
    \PackageWarningNoLine{glossaries}{##1}%
  }%
  \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
  \renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
\fi
}
\newcommand*\glsshowtarget}[1]{%
  \ifmmode
    \glsshowtargetinner{#1}%
  \else
    \ifinner
      \glsshowtargetinner{#1}%
    \else
      \glsshowtargetouter{#1}%
    \fi
  \fi
}
\newcommand*\glsshowtargetfonttext}[1]{%
  \ifmmode \nfss@text{\glsshowtargetfont #1}\else {\glsshowtargetfont #1}\fi
}
\newcommand*\glsshowtargetinner}[1]{\glsshowtargetfonttext{[#1]}}
\newcommand*\glsshowtargetouter}[1]{%
  \glsshowtargetsymbolsymbol\marginpar{\glsshowtargetsymbolsymbol\glsshowtargetfont #1}}
\newcommand*\glsshowtargetsymbolsymbol}{\tiny$\triangleright$}
\newcommand*\glsshowtargetfont}{\ttfamily\footnotesize}
\newcommand*\@glsshowtarget}[1]{%
\newcommand*\glsdoshowtarget}[2]{\@glsshowtarget{#1}#2}
\newrobustcmd*\@@glsshowtarget}[1]{%
  \begingroup
  \protected@edef\@gls@tmp{#1}%
  \@onelevel@sanitize\@gls@tmp
  \expandafter\glsshowtarget\expandafter{\@gls@tmp}%
  \endgroup
}
\newcommand*\@glsshowaccsupp}[3]{}
\newrobustcmd*\glsshowaccsupp}[3]{%
  \ifstrempy{#1}%
  {\@@glsshowtarget{/#2 (#3)}}%
  {\@@glsshowtarget{/#2 (#3) [#1]}}%
}

```

```

\newcommand*{\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries\space (or move
  \string\newglossaryentry\space
  definitions into the preamble)}}%
{You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key. It may
  be that the 'see' key has been written to the .glsdefs
  file from the previous run, in which case you need to
  move your definitions
  to the preamble if you don't want to use
  \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoindex}{%
  [\gls@seenoindex@val\gls@seenoindex@nr]{error,warn,ignore}{%
  \ifcase\gls@seenoindex@nr
    \renewcommand*{\@gls@see@noindex}{%
      \PackageError{glossaries}%
      {'\gls@xr@key' key may only be used after \string\makeglossaries\space
      or \string\makenoidxglossaries}%
      {You must use \string\makeglossaries\space
      or \string\makenoidxglossaries\space before defining
      any entries that have a '\gls@xr@key' key}%
    }%
  \or
    \renewcommand*{\@gls@see@noindex}{%
      \GlossariesWarning{'\gls@xr@key' key ignored}%
    }%
  \or
    \renewcommand*{\@gls@see@noindex}{}%
  \fi
}
\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}%
  {\newcommand*{\@@glossarysec}{section}}%
  {\newcommand*{\@@glossarysec}{chapter}}
\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
subsection,subsubsection,paragraph,subparagraph}[section]{%
  \renewcommand*{\@@glossarysec}{#1}}
\newcommand*{\@@glossarysecstar}{*}
\newcommand*{\@@glossaryseclabel}{}
\newcommand*{\glsautoprefix}{}
\define@choicekey{glossaries.sty}{numberedsection}%
  [\gls@numberedsection@val\gls@numberedsection@nr]{%
  false,nolabel,autolabel,nameref}[nolabel]{%
  \ifcase\gls@numberedsection@nr\relax

```

```

\renewcommand*\@@glossarysecstar}{*}%
\renewcommand*\@@glossaryseclabel}{}%
\or
\renewcommand*\@@glossarysecstar}{}%
\renewcommand*\@@glossaryseclabel}{}%
\or
\renewcommand*\@@glossarysecstar}{}%
\renewcommand*\@@glossaryseclabel}{%
\label{\glsautoprefix\@glo@type}}%
\or
\renewcommand*\@@glossarysecstar}{*}%
\renewcommand*\@@glossaryseclabel}{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}}%
\fi
}
\@ifpackageloaded{classicthesis}
{\newcommand*\@glossary@default@style}{index}}
{\newcommand*\@glossary@default@style}{list}}
\define@key{glossaries.sty}{style}{%
\def\@glossary@default@style{#1}%
}
\newcommand*\@gls@declareoption}[2]{%
\DeclareOptionX{#1}{#2}%
\DeclareOption{#1}{#2}%
}
\newcommand*\@glossaryentrynumbers}[1]{#1\@gls@save@numberlist{#1}}
\@gls@declareoption{nonumberlist}{%
\renewcommand*\@glossaryentrynumbers}[1]{\@gls@save@numberlist{#1}}%
}
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{%
\gls@savenumberlistfalse
\newcommand*\@glo@seeautonumberlist{}
\@gls@declareoption{seeautonumberlist}{%
\renewcommand*\@glo@seeautonumberlist}{%
\def\@glo@prefix{\glsnextpages}%
}%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{%
\gls@esclocationfalse
\newcommand*\@gls@loadlong{\RequirePackage{glossary-long}}
\@gls@declareoption{no-long}{\renewcommand*\@gls@loadlong}{}}
\IfFileExists{supertabular.sty}{%
\newcommand*\@gls@loadsuper{\RequirePackage{glossary-super}}}{%
\newcommand*\@gls@loadsuper}{}}
\@ifpackageloaded{xtab}{\renewcommand*\@gls@loadsuper}{}}{}
\@gls@declareoption{nosuper}{\renewcommand*\@gls@loadsuper}{}}{}
\newcommand*\@gls@loadlist{\RequirePackage{glossary-list}}
\@gls@declareoption{no-list}{%
\renewcommand*\@gls@loadlist}{%

```

```

\ifdefstring{\@glossary@default@style}{list}%
{\let\@glossary@default@style\relax}%
}%
}%
}
\newcommand*\@gls@loadtree{\RequirePackage{glossary-tree}}
\@gls@declareoption{notree}{\renewcommand*\@gls@loadtree{}}
\@gls@declareoption{nostyles}{%
\renewcommand*\@gls@loadlong}{}%
\renewcommand*\@gls@loadsuper}{}%
\renewcommand*\@gls@loadlist}{}%
\renewcommand*\@gls@loadtree}{}%
\let\@glossary@default@style\relax
}
\newcommand*\glspostdescription{%
\ifglsnopostdot\else.\spacefactor\sfcode'\. \fi
}
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{}
\glsnopostdotfalse
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{}
\glsnogroupskipfalse
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{}
\@ifclassloaded{memoir}
{%
\glsucmarktrue
}%
{%
\glsucmarkfalse
}
\newcommand*\@gls@define@glossaryentrycounter{%
\ifglsentrycounter
\ifundef\c@glossaryentry
{%
\ifx\@gls@counterwithin\@empty
\newcounter{glossaryentry}%
\else
\newcounter{glossaryentry}[\@gls@counterwithin]%
\fi
\def\theHglossaryentry{\currentglossary.\theglossaryentry}%
}%
}%
\fi
}
\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\glsentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
\renewcommand*\@gls@counterwithin{#1}%
\glsentrycountertrue
\@gls@define@glossaryentrycounter
}
}

```



```

\newcommand*{\@gls@counterwithin}{}
\newcommand{\@gls@define@glossarysubentrycounter}{%
  \ifundef\c@glossarysubentry
  {%
    \ifglssubentrycounter
    \ifglsubentrycounter
    \newcounter{glossarysubentry}[glossaryentry]%
    \else
    \newcounter{glossarysubentry}%
    \fi
    \def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
  \fi
  }%
}%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
\glssubentrycounterfalse
\newcommand{\glsindexingsetting}{\ifglsubentrycounter\else makeindex\fi}
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none,clear}{%
  \renewcommand*{\@glo@default@sorttype}{#1}%
  \csname @gls@setupsort@#1\endcsname
}
\newcommand*{\glsprestandardsort}[3]{%
  \glsdosanitizesort
}
\newcommand*{\@glo@check@sortallowed}[1]{}
\newcommand*{\@gls@setupsort@standard}{%
  \def\do@glo@storeentry{\@glo@storeentry}%
  \def\@gls@defsortcount##1{}%
  \def\@gls@defsort##1##2{%
    \ifx\@glo@sort\@glsdefaultsort
    \let\@glo@sort\@glo@name
    \fi
    \let\glsdosanitizesort\@gls@sanitizesort
    \glsprestandardsort{\@glo@sort}{##1}{##2}%
    \expandafter\protected\xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
  }%
  \def\@gls@setsort##1{}%
  \let\@glo@check@sortallowed\@gobble
}
\@gls@setupsort@standard
\newcommand*\glsortnumberfmt[1]{%
  \ifnum#1<100000 0\fi
  \ifnum#1<10000 0\fi
  \ifnum#1<1000 0\fi
  \ifnum#1<100 0\fi
  \ifnum#1<10 0\fi
  \number#1%
}

```

```

\newcommand*{\@gls@setupsort@def}{%
  \def\do@glo@storeentry{\@glo@storeentry}%
  \def\@gls@defsortcount##1{%
    \expandafter\global
    \expandafter\newcount\csname glossary@##1@sortcount\endcsname
  }%
  \def\@gls@defsort##1##2{%
    \ifcsundef{glossary@##1@sortcount}%
    {\@gls@defsortcount{##1}}%
    {}%
    \expandafter\global\expandafter
    \advance\csname glossary@##1@sortcount\endcsname by 1\relax
    \expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
      \expandafter\glssortnumberfmt
      {\csname glossary@##1@sortcount\endcsname}}%
  }%
  \def\@gls@setsort##1{%
    \let\@glo@check@sortallowed\@gobble
  }
\newcommand*{\@gls@setupsort@use}{%
  \let\do@glo@storeentry\@gobble
  \def\@gls@defsortcount##1{%
    \expandafter\global
    \expandafter\newcount\csname glossary@##1@sortcount\endcsname
  }%
  \def\@gls@defsort##1##2{%
    \expandafter\gdef\csname glo@##2@sort\endcsname{}%
  }%
  \def\@gls@setsort##1{%
    \protected@edef\@glo@parent{\csname glo@##1@parent\endcsname}%
    \ifx\@glo@parent\@empty
    \else
      \expandafter\@gls@setsort\expandafter{\@glo@parent}%
    \fi
    \protected@edef\@glo@type{\csname glo@##1@type\endcsname}%
    \protected@edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
    \ifx\@gls@tmp\@empty
      \expandafter\global\expandafter
      \advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
      \expandafter\protected@xdef\csname glo@##1@sort\endcsname{%
        \expandafter\glssortnumberfmt
        {\csname glossary@\@glo@type @sortcount\endcsname}}%
      \@glo@storeentry{##1}%
    \fi
  }%
  \let\@glo@check@sortallowed\@gobble
}
\newcommand*{\@gls@setupsort@none}{%
  \def\glsindexingsetting{none}%
  \def\do@glo@storeentry##1{%

```

```

\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
  \expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort
}%
\def\@gls@setsort##1{%
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
{Option sort=none not allowed with \string##1}%
{(Use sort=def instead)}}%
}
\newcommand*\@gls@setupsort@clear{%
\def\glsindexingsetting{none}%
\def\do@glo@storeentry##1{%
\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{\csgdef{glo@##2@sort}{}}%
\def\@gls@setsort##1{%
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
{Option sort=clear not allowed with \string##1}%
{(Use sort=def instead)}}%
}
\newcommand*\glsdefmain{%
\if@gls@docloaded
\newglossary[glg2]{main}{gls2}{glo2}{\glossaryname}%
\else
\newglossary{main}{gls}{glo}{\glossaryname}%
\fi
\newcommand*\gls@tr@set@main@toctitle{%
\translatelet{\glossarytoctitle}{Glossary}%
}%
}
\newcommand*\glsdefaulttype{main}
\newcommand*\acronymtype{\glsdefaulttype}
\@gls@declareoption{nomain}{%
\let\glsdefaulttype\relax
\renewcommand*\glsdefmain}{%
}
\define@boolkey{glossaries.sty}[gls]{acronym}[true]{%
\ifglsacronym
\renewcommand*\@gls@do@acronymsdef{%
\DeclareAcronymList{acronym}%
\newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
\renewcommand*\acronymtype{acronym}%
\newcommand*\gls@tr@set@acronym@toctitle}{%
\translatelet{\glossarytoctitle}{Acronyms}%
}%
}%
\else
\let\@gls@do@acronymsdef\relax
\fi
}
\AtBeginDocument{%

```

```

\ifglsacronym
  \providecommand*\printacronyms[1][1]{%
    \printglossary[type=\acronymtype,#1]%
  }
}
\newcommand*\@gls@do@acronymsdef{}
\@gls@declareoption{acronyms}{%
  \glsacronymtrue
  \def\@gls@do@acronymsdef{%
    \DeclareAcronymList{acronym}%
    \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
    \renewcommand*\@acronymtype{acronym}%
    \newcommand*\gls@tr@set@acronym@toctitle{%
      \translatelet{\glossarytoctitle}{Acronyms}%
    }%
  }%
}
\newcommand*\@gls@acronymlists{}
\newcommand*\@addtoacronymlists[1]{%
  \ifx\@gls@acronymlists\@empty
    \protected@xdef\@gls@acronymlists{#1}%
  \else
    \protected@xdef\@gls@acronymlists{\@gls@acronymlists,#1}%
  \fi
}
\newcommand*\DeclareAcronymList[1]{%
  \glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}
\newcommand{\glsIfListOfAcronyms}[1]{%
  \protected@edef\@do@gls@islistofacronyms{%
    \noexpand\@gls@islistofacronyms{#1}{\@gls@acronymlists}}%
  \@do@gls@islistofacronyms
}
\newcommand{\@gls@islistofacronyms}[4]{%
  \def\gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%
    \def\@gls@before{##1}\def\@gls@after{##2}}%
  \gls@islistofacronyms,#2,#1,\@nil\end@gls@islistofacronyms
  \ifx\@gls@after\@nnil
    #4%
  \else
    #3%
  \fi
}
\newif\if@gls@isacronymlist
\newcommand*\@gls@checkisacronymlist[1]{%
  \glsIfListOfAcronyms{#1}%
  {\@gls@isacronymlisttrue}{\@gls@isacronymlistfalse}%
}
\newcommand*\SetAcronymLists[1]{%
  \renewcommand*\@gls@acronymlists{#1}%
}

```

```

}
\define@key{glossaries.sty}{acronymlists}{%
  \DeclareAcronymList{#1}%
}
\newcommand{\glscounter}{page}
\define@key{glossaries.sty}{counter}{%
  \renewcommand*{\glscounter}{#1}%
}
\newcommand*{\@gls@nohyperlist}{}
\newcommand*{\GlsDeclareNoHyperList}[1]{%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*{\@gls@nohyperlist}{#1}%
  }%
  {%
    \appto\@gls@nohyperlist{,#1}%
  }%
}
\define@key{glossaries.sty}{nohypertypes}{%
  \GlsDeclareNoHyperList{#1}%
}
\newcommand*{\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{#1}%
}
\newcommand*{\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{#1}%
}
\newcommand{\glosortentrieswarning}{%
  \typeout{Using TeX to sort glossary entries---this may
  take a while}%
}
\@gls@declareoption{nowarn}{%
  \if@gls@debug
    \GlossariesWarning{Warnings can't be suppressed in debug mode}%
  \else
    \renewcommand*{\GlossariesWarning}[1]{}%
    \renewcommand*{\GlossariesWarningNoLine}[1]{}%
    \renewcommand*{\glosortentrieswarning}{}%
    \renewcommand*{\@gls@missinglang@warn}[2]{}%
  \fi
}
\newcommand*{\@gls@missinglang@warn}[2]{%
  \PackageWarningNoLine{glossaries}%
  {No language module detected for '#1'.\MessageBreak
  Language modules need to be installed separately.\MessageBreak
  Please check on CTAN for a bundle called\MessageBreak
  'glossaries-#2' or similar}%
}
\@gls@declareoption{nolangwarn}{%
  \renewcommand*{\@gls@missinglang@warn}[2]{}%
}

```

```

}
\newcommand*{\@gls@warnonglossdefined}{%
  \GlossariesWarning{Overriding \string\printglossary}%
}
\newcommand*{\@gls@warnontheGLOSSdefined}{%
  \GlossariesWarning{Overriding 'theGLOSS' environment}%
}
\@gls@declareoption{noredefwarn}{%
  \renewcommand*{\@gls@warnonglossdefined}{}%
  \renewcommand*{\@gls@warnontheGLOSSdefined}{}%
}
\newcommand*{\glssetexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@expand@field{##1}{#1}{##2}%
  }%
}
\newcommand*{\glssetnoexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@noexpand@field{##1}{#1}{##2}%
  }%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}
\glssetnoexpandfield{descplural}
\newcommand*{\@gls@sanitizename}{}
\glssetnoexpandfield{name}
\newcommand*{\@gls@sanitizesymbol}{}
\glssetnoexpandfield{symbol}
\glssetnoexpandfield{symbolplural}
\newcommand*{\@gls@sanitizesort}{%
  \ifgls@sanitizesort
    \@gls@sanitizesort
  \else
    \@gls@nosanitizesort
  \fi
}
\newcommand*\@gls@sanitizesort{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*\@gls@nosanitizesort{}
\newcommand*\@gls@noidx@sanitizesort{%
  \ifdefvoid\@glo@sort
  {}%
  {%
    \expandafter\@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
  \def\@glo@sort{#1#2}%
  \@onelevel@sanitize\@glo@sort
}

```

```

}
\newcommand*{\@@gls@noidx@nosanitizesort}{%
  \ifdefvoid\@glo@sort
  {}%
  {%
    \expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
  \bgroup
  \glsnoidxstripaccents
  \protected@xdef\@@glo@sort{#1#2}%
  \egroup
  \let\@glo@sort\@@glo@sort
}
\newcommand*\glsnoidxstripaccents{%
  \let\IeC\@firstofone
  \let\add@accent@\@secondoftwo
  \let\@text@composite@x@\@secondoftwo
  \let\@tabacckludge\@secondoftwo
  \expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
  \let'\@firstofone
  \let'\@firstofone
  \let\~\@firstofone
  \let"\@firstofone
  \let\u\@firstofone
  \let\t\@firstofone
  \let\d\@firstofone
  \let\r\@firstofone
  \let=\@firstofone
  \let.\@firstofone
  \let~\@firstofone
  \let\v\@firstofone
  \let\H\@firstofone
  \let\c\@firstofone
  \let\b\@firstofone
  \let\a\@secondoftwo
  \def\AE{AE}%
  \def\ae{ae}%
  \def\OE{OE}%
  \def\oe{oe}%
  \def\AA{AA}%
  \def\aa{aa}%
  \def\L{L}%
  \def\l{l}%
  \def\O{O}%
  \def\o{o}%

```

```

\def\SS{SS}%
\def\ss{ss}%
\def\th{th}%
\def\TH{TH}%
\def\dh{dh}%
\def\DH{DH}%
}
\@ifl@t@r\fmtversion{2019/10/01}
{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}
\newcommand{\@gls@removedoption}[1]{%
\PackageError{glossaries}{obsolete package option ‘#1’ has been
removed. Rollback required or use a newer alternative}
{Try \string\usepackage[#1]{glossaries}[=v4.46]}%
}
\define@boolkey[gls]{sanitize}{description}[true]{%
\GlossariesWarning{sanitize={description} package option deprecated}%
\ifgls@sanitize@description
\glssetnoexpandfield{desc}%
\glssetnoexpandfield{descplural}%
\else
\glssetexpandfield{desc}%
\glssetexpandfield{descplural}%
\fi
}
\define@boolkey[gls]{sanitize}{name}[true]{%
\GlossariesWarning{sanitize={name} package option deprecated}%
\ifgls@sanitize@name
\glssetnoexpandfield{name}%
\else
\glssetexpandfield{name}%
\fi
}
\define@boolkey[gls]{sanitize}{symbol}[true]{%
\GlossariesWarning{sanitize={symbol} package option deprecated}%
\ifgls@sanitize@symbol
\glssetnoexpandfield{symbol}%
\glssetnoexpandfield{symbolplural}%
\else
\glssetexpandfield{symbol}%
\glssetexpandfield{symbolplural}%
\fi
}
\define@boolkey{glossaries.sty}[gls]{sanitizesort}[true]{%
\ifglssanitizesort
\glssetnoexpandfield{sortvalue}%
\renewcommand*{\@gls@noidx@setsanitizesort}{%
\glssanitizesorttrue

```



```

        \glsssetnoexpandfield{sortvalue}%
    }%
\else
    \glsssetexpandfield{sortvalue}%
    \renewcommand*{\@gls@noidx@setsanitizesort}{%
        \glsssanitizesortfalse
        \glsssetexpandfield{sortvalue}%
    }%
\fi
}
\glsssanitizesorttrue
\glsssetnoexpandfield{sortvalue}%
\newcommand*{\@gls@noidx@setsanitizesort}{%
    \glsssanitizesortfalse
    \glsssetexpandfield{sortvalue}%
}
\define@choicekey[gls]{sanitize}{sort}{true,false}[true]{%
    \setbool{glsssanitizesort}{#1}%
    \ifglsssanitizesort
        \glsssetnoexpandfield{sortvalue}%
    \else
        \glsssetexpandfield{sortvalue}%
    \fi
    \GlossariesWarning{sanitize={sort} package option
        deprecated. Use sanitizesort instead}%
}
\define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
    \ifthenelse{\equal{#1}{none}}{%
        {%
            \GlossariesWarning{sanitize package option deprecated}%
            \glsssetexpandfield{name}%
            \glsssetexpandfield{symbol}%
            \glsssetexpandfield{symbolplural}%
            \glsssetexpandfield{desc}%
            \glsssetexpandfield{descplural}%
        }%
    }%
    \setkeys[gls]{sanitize}{#1}%
}%
}
\newif\ifglstranslate
\newcommand*\@gls@usetranslator{%
    \@ifpackageloaded{polyglossia}%
    {%
        \let\glsifusetranslator\@secondoftwo
    }%
    {%
        \@ifpackageloaded{babel}%
        {%
            \IfFileExists{translator.sty}%

```

```

        {%
        \RequirePackage{translator}%
        \let\glsifusetranslator\@firstoftwo
        }%
        {%
    }%
    {%
}
}
\newcommand{\glsifusedtranslatordict}[3]{%
  \glsifusetranslator
  {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
  {#3}%
}
\@gls@declareoption{nottranslate}{%
  \glstranslatefalse
  \let\@gls@usetranslator\relax
  \let\glsifusetranslator\@secondoftwo
}
\define@choicekey{glossaries.sty}{translate}{%
  [\gls@translate@val\gls@translate@nr]%
  {true,false,babel}[true]%
  {%
    \ifcase\gls@translate@nr\relax
    \glstranslatetrue
    \renewcommand*\@gls@usetranslator{%
      \@ifpackageloaded{polyglossia}%
      {%
        \let\glsifusetranslator\@secondoftwo
      }%
      {%
        \@ifpackageloaded{babel}%
        {%
          \IfFileExists{translator.sty}%
          {%
            \RequirePackage{translator}%
            \let\glsifusetranslator\@firstoftwo
          }%
          {}%
        }%
        {}%
      }%
    }%
  }%
  \or
  \glstranslatefalse
  \let\@gls@usetranslator\relax
  \let\glsifusetranslator\@secondoftwo
  \or
  \glstranslatetrue
  \let\@gls@usetranslator\relax

```

```

        \let\glsifusetranslator\@secondoftwo
    \fi
}
\glstranslatefalse
\let\glsifusetranslator\@secondoftwo
\@ifpackageloaded{translator}%
{%
    \glstranslatetrue
    \let\glsifusetranslator\@firstoftwo
}%
{%
    \@for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
    {
        \@ifpackageloaded{\gls@thissty}%
        {%
            \glstranslatetrue
            \@endfortrue
        }%
        {}%
    }
}
\newcommand{\@gls@requested@languages}{}
\define@key{glossaries.sty}{languages}{%
    \renewcommand{\@gls@requested@languages}{#1}%
    \glstranslatetrue
    \let\@gls@usetranslator\relax
    \let\glsifusetranslator\@secondoftwo
}
\define@boolkey{glossaries.sty}[gls]{indexonlyfirst}[true]{}
\glsindexonlyfirstfalse
\define@boolkey{glossaries.sty}[gls]{hyperfirst}[true]{}
\glshyperfirsttrue
\newcommand*{\@gls@setacrstyle}{}
\define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{}
\ifglsacrfootnote
    \@gls@removedoption{footnote}%
\fi
}
\define@boolkey{glossaries.sty}[glsacr]{description}[true]{}
\ifglsacrdescription
    \@gls@removedoption{description}%
\fi
}
\define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{}
\ifglsacrsmallcaps
    \@gls@removedoption{smallcaps}%
\fi
}
\define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{}
\ifglsacrsmaller

```

```

    \@gls@removedoption{smaller}%
  \fi
}
\define@boolkey{glossaries.sty}[glsacr]{dua}[true]{%
  \ifglsacrdua
    \@gls@removedoption{dua}%
  \fi
}
\define@boolkey{glossaries.sty}[glsacr]{shortcuts}[true]{}
\newcommand*\@glsorder{word}
\newcommand*\@glsorder[1]{}
\define@choicekey{glossaries.sty}{order}{word,letter}{%
  \def\glsorder{#1}}
\newif\ifglsxindy
\glsxindyfalse
\@gls@declareoption{makeindex}{\glsxindyfalse}
\define@boolkey[gls]{xindy}{glsnumbers}[true]{}
\gls@xindy@glsnumberstrue
\def\@xdy@main@language{\language}%
\newcommand*\@gls@xdy@set@lang[1]{%
  \ifcsundef{\@xdy@#1@language}%
    {%
      \ifcsundef{\@xdy@\glsdefaulttype @language}
        {\let\@gls@langmod\@xdy@main@language}%
        {\letcs\@gls@langmod{\@xdy@\glsdefaulttype @language}}%
      }%
    {\letcs\@gls@langmod{\@xdy@#1@language}}%
  }
}
\define@key[gls]{xindy}{language}{\def\@xdy@main@language{#1}}
\ifvoid\inputencodingname
  {\def\gls@codepage{utf8}}%
  {\def\gls@codepage{\inputencodingname}}
\define@key[gls]{xindy}{codepage}{\def\gls@codepage{#1}}
\define@key{glossaries.sty}{xindy}[][%
  \glsxindytrue
  \setkeys[gls]{xindy}{#1}%
}
\@gls@declareoption{xindygloss}{%
  \glsxindytrue
}
\@gls@declareoption{xindynoglsnumbers}{%
  \glsxindytrue
  \gls@xindy@glsnumbersfalse
}
\providecommand{\@domakeglossaries}[1]{\if@filesw #1\fi}
\@gls@declareoption{disablemakegloss}{%
  \ifdequal\makeglossaries\@no@makeglossaries
    {%
      \GlossariesWarning{Option 'disablemakegloss' has no effect
        (\string\makenoidxglossaries\space has already been used)}%
    }
  }

```

```

}%
{%
\ifdefequal\@makeglossary\@gobble
{%
\GlossariesWarning{Option 'disablemakegloss' has no effect
(\string\makeglossaries\space has already been used)}%
}%
{%
\renewcommand{\@domakeglossaries}[1]{%
\PackageInfo{glossaries}{\string\makeglossaries\space and
\string\makenoidxglossaries\space have been disabled}%
}%
}%
}%
}
\@gls@declareoption{restoremakegloss}{%
\ifdefequal\makeglossaries\@no@makeglossaries
{%
\GlossariesWarning{Option 'restoremakegloss' has no effect
(\string\makenoidxglossaries\space has already been used)}%
}%
{%
\ifdefequal\@makeglossary\@gobble
{%
\GlossariesWarning{Option 'restoremakegloss' has no effect
(\string\makeglossaries\space has already been used)}%
}%
{%
\PackageInfo{glossaries}{\string\makeglossaries\space and
\string\makenoidxglossaries\space have been restored}%
\let\@domakeglossaries\@firstofone
}%
}%
}
}
\newcommand*\@do@write@glslabels{%
\AtEndDocument{\@do@write@glslabels}%
\let\@do@write@glslabels\relax
}
\newcommand*\@do@write@glslabels{%
\newwrite\@gls@labelsfile
\immediate\openout\@gls@labelsfile=\jobname.glslabels
\protected@edef\@gls@fulllist{\@gls@types}%
\ifdefempty{\@ignored@glossaries}{%
{\protected@eappto\@gls@fulllist{\@ignored@glossaries}}%
\forallglsentries[\@gls@fulllist]{\@glsentry}%
}%
\ifdefempty{\@glsentry}{%
{%
\ifx\do@write@glslabelsname\@gobble
\def\@gls@thisentryname{%

```

```

        \else
            \letcs{\@gls@thisentryname}{glo@\glsdetoklabel{\@glsentry}@name}%
            \@onelevel@sanitize\@gls@thisentryname
        \fi
        \immediate\write\@gls@labelsfile
            {\@glsentry\do@write@glslabelsname{\@gls@thisentryname}}%
    }%
}%
\immediate\closeout\@gls@labelsfile
}
\newcommand*\do@write@glslabelsname[1]{
\begingroup
\catcode'\^^I12
\gdef\@write@glslabelsname#1{\^^I#1}
\endgroup
\@gls@declareoption{writeglslabels}{\do@write@glslabels
\let\do@write@glslabelsname\gobble}
\@gls@declareoption{writeglslabelnames}{\do@write@glslabels
\let\do@write@glslabelsname\write@glslabelsname}
\newif\ifglsautomake
\newcommand{\gls@automake@nr}{1}
\newcommand{\gls@automake@makegloss}{}
\newcommand{\gls@automake@option}{automake=\gls@automake@val}
\newcommand*\@gls@do@automake@err}{%
\PackageError{glossaries}{You must use
\string\makeglossaries\space with \gls@automake@option}
{%
    Either remove the \gls@automake@option\space setting or
    add \string\makeglossaries\space to your document preamble.%
}%
}%
}%
\define@choicekey{glossaries.sty}{automake}%
[\gls@automake@val\gls@automake@nr]{true,false,immediate,makegloss,lite,delayed}[immediate]{%
\ifcase\gls@automake@nr\relax
\GlossariesWarning{automake=true deprecated. Use automake=delayed for a delayed write}%
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomakefalse
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{makeglossaries}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
\or

```

```

\def\gls@automake@nr{0}%
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\fi
\renewcommand{\gls@automake@option}{automake=\gls@automake@val}%
\ifglsautomake
\let\@gls@doautomake\@gls@do@automake@err
\else
\renewcommand*{\@gls@doautomake}{}%
\fi
}
\glsautomakefalse
\newcommand*{\@gls@doautomake}{}
\AtEndDocument{\@gls@doautomake}
\@gls@declareoption{automakegloss}{%
\glsautomaketrue
\renewcommand{\gls@automake@option}{automakegloss}%
\def\gls@automake@val{makegloss}%
\def\gls@automake@nr{3}%
\renewcommand{\gls@automake@makegloss}{makeglossaries}%
\let\@gls@doautomake\@gls@do@automake@err
}
\@gls@declareoption{automakeglosslite}{%
\glsautomaketrue
\renewcommand{\gls@automake@option}{automakeglosslite}%
\def\gls@automake@val{lite}%
\def\gls@automake@nr{4}%
\renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
\let\@gls@doautomake\@gls@do@automake@err
}
}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
\ifgls savewrites
\renewcommand*{\glswritefiles}{\@glswritefiles}%
\else
\let\glswritefiles\@empty
\fi
}
\gls savewritesfalse
\let\glswritefiles\@empty
\define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{%
\ifbool{glscompatible-3.07}%
{%
\@gls@removedoption{compatible-3.07}%
}%
}%
}
\boolfalse{glscompatible-3.07}
\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
\ifbool{glscompatible-2.07}%
{%

```

```

    \@gls@removedoption{compatible-2.07}%
  }%
  {}%
}
\boolfalse{glscompatible-2.07}
\let\gls@original@makeglossary\makeglossary
\let\gls@original@glossary\glossary
\newcommand*{\@gls@override@glossary}[1][main]{%
  \GlossariesWarning{Use of \string\glossary\space with
glossaries.sty is deprecated. \MessageBreak Indexing should be performed
with the user level \MessageBreak commands, such as \string\gls\space or
\string\glsadd. If you need the \MessageBreak original definition of
\string\glossary\space use the package \MessageBreak options
kernelglossredefs=false (to restore the \MessageBreak former definition of
\string\glossary) and nomain (if the \MessageBreak file extensions cause a
conflict)}%
  \gls@glossary{#1}%
}
\define@choicekey{glossaries.sty}{kernelglossredefs}%
[\@gls@kernelglossredefs@val\@gls@kernelglossredefs@nr]{true,false,nowarn}[true]%
{%
  \ifcase\@gls@kernelglossredefs@nr\relax
  \def\glossary{\@gls@override@glossary}%
  \def\makeglossary{%
    \GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is deprecated. Use \string\makeglossaries\space
instead. If you need the original definition of
\string\makeglossary\space use the package options
kernelglossredefs=false (to prevent redefinition of
\string\makeglossary) and nomain (if the file extensions cause a
conflict)}%
    \makeglossaries
  }%
  \or
  \let\glossary\gls@original@glossary
  \let\makeglossary\gls@original@makeglossary
  \or
  \def\makeglossary{\makeglossaries}%
  \renewcommand*{\@gls@override@glossary}[1][main]{%
    \gls@glossary{##1}%
  }%
  \fi
}
\@gls@declareoption{symbols}{%
  \let\@gls@do@symbolsdef\@gls@symbolsdef
}
\newcommand*{\@gls@do@symbolsdef}{%
\newcommand*{\@gls@symbolsdef}{%
  \newglossary[slg]{symbols}{sls}{slo}{\glssymbolsgroupname}%
  \newcommand*{\printsymbols}[1][\]{\printglossary[type=symbols,##1]}%
}

```



```

\newcommand*\gls@tr@set@symbols@toctitle}{%
  \translatelet{\glossarytoctitle}{Symbols (glossaries)}%
}%
}%
\@gls@declareoption{numbers}{%
  \let\@gls@do@numbersdef\@gls@numbersdef
}
\newcommand*\@gls@do@numbersdef{}
\newcommand*\@gls@numbersdef{%
  \newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
  \newcommand*\@printnumbers}[1][\]{\printglossary[type=numbers,##1]}%
  \newcommand*\gls@tr@set@numbers@toctitle}{%
    \translatelet{\glossarytoctitle}{Numbers (glossaries)}%
  }%
}%
\@gls@declareoption{index}{%
  \ifx\@gls@do@indexdef\@empty
    \let\@gls@do@indexdef\@gls@indexdef
  \fi
}
\@gls@declareoption{noglossaryindex}{%
  \let\@gls@do@indexdef\relax
}
\newcommand*\@gls@do@indexdef{}
\newcommand*\@gls@indexdef{%
  \newglossary[ilg]{index}{ind}{idx}{\indexname}%
  \newcommand*\@printindex}[1][\]{\printglossary[type=index,##1]}%
  \newcommand*\@newterm}[2][\]{%
    \newglossaryentry{##2}%
    {type={index},name={##2},description={\nopostdesc},##1}}
  \let\@gls@do@indexdef\relax
}%
\define@choicekey{glossaries.sty}{mfirstuc}{%
  [\gls@mfirstuc@val\gls@mfirstuc@nr]%
  {expanded,unexpanded}%
  {%
    \ifcase\gls@mfirstuc@nr
      % expanded
      \renewcommand*\@gls@makefirstuc}[1]{\MFUsentencecase{##1}}%
    \or
      % unexpanded
      \renewcommand*\@gls@makefirstuc}[1]{\MFUsentencecase{\unexpanded{##1}}}%
    \fi
  }
}
\@for\CurrentOption :=\@declaredoptions\do{%
  \ifx\CurrentOption\@empty
  \else
    \@expandtwoargs
    \in@ {\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
  \ifin@

```

```

        \@use@ption
        \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
    \fi
\fi
}
\ProcessOptionsX
\ifglsautomake
\ifdef\directlua{\RequirePackage{shellesc}}{}
\fi
\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,notranslate,languages,nolong,nosuper,notree,nostyles,%
nomain,noglossaryindex}
\newcommand*{\setupglossaries}[1]{%
\renewcommand*{\@gls@setacrstyle}{}%
\ifglsacrshortcuts
\def\@gls@setupshortcuts{\glsacrshortcutstrue}%
\else
\def\@gls@setupshortcuts{%
\ifglsacrshortcuts
\DefineAcronymSynonyms
\fi
}%
\fi
\glsacrshortcutsfalse
\let\@gls@do@numbersdef\relax
\let\@gls@do@symbolssdef\relax
\let\@gls@do@indexdef\relax
\let\@gls@do@acronymsdef\relax
\ifglsentrycounter
\let\@gls@doentrycounterdef\relax
\else
\let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
\fi
\ifglssubentrycounter
\let\@gls@dosubentrycounterdef\relax
\else
\let\@gls@dosubentrycounterdef\@gls@define@glossarysubentrycounter
\fi
\setkeys{glossaries.sty}{#1}%
\@gls@setacrstyle
\@gls@setupshortcuts
\@gls@do@acronymsdef
\@gls@do@numbersdef
\@gls@do@symbolssdef
\@gls@do@indexdef
\@gls@doentrycounterdef
\@gls@dosubentrycounterdef
}
\ifthenelse{\equal{\glscounter}{section}}{%

```

```

{%
  \ifcsundef{chapter}{}%
  {%
    \let\@gls@old@chapter\@chapter
    \def\@chapter[#1]#2{\@gls@old@chapter[#1]}{#2}%
    \ifcsundef{hyperdef}{\hyperdef{section}{\thesection}}{}%
  }%
}%
{}
\newcommand*\@gls@onlypremakeg{}
\newcommand*\@onlypremakeg[1]{%
  \ifx\@gls@onlypremakeg\@empty
    \def\@gls@onlypremakeg{#1}%
  \else
    \expandafter\toks@\expandafter{\@gls@onlypremakeg}%
    \edef\@gls@onlypremakeg{\the\toks@,\noexpand#1}%
  \fi
}
\newcommand*\@disable@onlypremakeg{%
\@for\@thiscs:=\@gls@onlypremakeg\do{%
  \expandafter\@disable@premakecs\@thiscs%
}}
\newcommand*\@disable@premakecs[1]{%
  \def#1{\PackageError{glossaries}{\string#1\space may only be
used before \string\makeglossaries}{You can't use
\string#1\space after \string\makeglossaries}}%
}
\providecommand*\@glossaryname{Glossary}
\providecommand*\@acronymname{Acronyms}
\newcommand*\@glssettoctitle[1]{%
  \def@glossarytoctitle{\csname @glotype@#1@title\endcsname}}
\providecommand*\@entryname{Notation}
\providecommand*\@descriptionname{Description}
\providecommand*\@symbolname{Symbol}
\providecommand*\@pagelistname{Page List}
\providecommand*\@glssymbolsgroupname{Symbols}
\providecommand*\@glsnumbersgroupname{Numbers}
\newcommand*\@glspluralsuffix{s}
\newcommand*\@glsacrpluralsuffix{\@glspluralsuffix}
\newcommand*\@glsupacrpluralsuffix{\@glstextup{\@glsacrpluralsuffix}}
\providecommand*\@seename{see}
\providecommand*\@andname{\&}
\newcommand*\@RequireGlossariesLang[1]{%
  \@ifundefined{ver@glossaries-#1.ldf}{\input{glossaries-#1.ldf}}{}%
}
\newcommand*\@ProvidesGlossariesLang[1]{%
  \ProvidesFile{glossaries-#1.ldf}%
}
\newcommand*\@addglossarytocaptions[1]{}
\ifglstranslate

```

```

\RequirePackage{tracklang}[2019/11/30]
\ifvoid\@gls@requested@languages{%
{%
  \for\@gls@thislang:=\@gls@requested@languages\do{%
    \TrackIfKnownLanguage{\@gls@thislang}{}%
    {\GlossariesWarning{unknown language ‘\@gls@thislang’}}%
  }%
}%
\@gls@usetranslator
\@ifpackageloaded{translator}
{%
  \ifboolexpr
  {
    test {\ifdefstring{\trans@languages}{English}}
    and not
    test {\ifdefstring{bbl@loaded}{english}}
  }
  {%
    \let\glsifusetranslator\@secondoftwo
  }%
  {%
    \usedictionary{glossaries-dictionary}%
    \renewcommand*{\addglossarytocaptions}[1]{%
      \ifcsundef{captions#1}{}%
      {%
        \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
        \expandafter\toks@\expandafter{\@gls@tmp
          \renewcommand*{\glossaryname}{\translate{Glossary}}}%
        }%
        \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
      }%
    }%
  }%
}%
\}
\}
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
  }%
}%
\}
\}

```

```

\glsifusetranslator
{%
  \renewcommand*\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
      {%
        \csuse{gls@tr@set@#1@toctitle}%
      }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*\glossaryname{\translate{Glossary}}%
  \renewcommand*\acronymname{\translate{Acronyms}}%
  \renewcommand*\entryname{\translate{Notation (glossaries)}}%
  \renewcommand*\descriptionname{%
    \translate{Description (glossaries)}}%
  \renewcommand*\symbolname{\translate{Symbol (glossaries)}}%
  \renewcommand*\pagelistname{%
    \translate{Page List (glossaries)}}%
  \renewcommand*\glsymbolsgroupname{%
    \translate{Symbols (glossaries)}}%
  \renewcommand*\glsnumbersgroupname{%
    \translate{Numbers (glossaries)}}%
  }{%
\fi
\DeclareRobustCommand*\nopostdesc{}
\newcommand*\@nopostdesc{%
  \let\org@glspostdescription\glspostdescription
  \def\glspostdescription{%
    \let\glspostdescription\org@glspostdescription}%
}
\newcommand*\@no@post@desc{\nopostdesc}
\newcommand*\glspar{\par}
\newcommand*\setStyleFile[1]{%
  \renewcommand*\gls@istfilebase}{#1}%
  \ifglxindy
    \def\istfilename{\gls@istfilebase.xdy}
  \else
    \def\istfilename{\gls@istfilebase.ist}
  \fi
}
\@onlypremakeg\setStyleFile
\ifglxindy
  \def\istfilename{\gls@istfilebase.xdy}
\else
  \def\istfilename{\gls@istfilebase.ist}
\fi
\newcommand*\gls@istfilebase{\jobname}
\newcommand*\@istfilename[1]{%
\newcommand*\glscompositor}{.}

```

```

\newcommand*\glsSetCompositor}[1]{%
  \renewcommand*\glscompositor{#1}}
\@onlypremakeg\glsSetCompositor
\newcommand*\@glsAlphacompositor{\glscompositor}
\ifglsxindy
  \newcommand*\glsSetAlphaCompositor[1]{%
    \renewcommand*\@glsAlphacompositor{#1}}
\else
  \newcommand*\glsSetAlphaCompositor[1]{%
    \glsnoxywarning\glsSetAlphaCompositor}
\fi
\@onlypremakeg\glsSetAlphaCompositor
\newcommand*\gls@suffixF{}
\newcommand*\glsSetSuffixF}[1]{%
  \renewcommand*\gls@suffixF{#1}}
\@onlypremakeg\glsSetSuffixF
\newcommand*\gls@suffixFF{}
\newcommand*\glsSetSuffixFF}[1]{%
  \renewcommand*\gls@suffixFF{#1}}
}
\ifcsundef{hyperlink}%
{%
  \newcommand*\glsnumberformat}[1]{#1}%
}%
{%
  \newcommand*\glsnumberformat}[1]{\glshypernumber{#1}}%
}
\newcommand{\glslocationcstoencap}[3]{\csuse{#2}{\csuse{#1}{#3}}}
\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*\glossarypreamble{%
  \csuse{@glossarypreamble@\currentglossary}}
}
\newcommand{\setglossarypreamble}[2][\glsdefaultttype]{%
  \ifglossaryexists*{#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*\glossarypostamble{}
\newcommand*\glossarysection}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@pglossarysection{#1}{#2}%
  }%
  \gls glossarymark{\glossarytoctitle}%
}

```

```

\ifcsundef{glossarymark}%
{%
  \newcommand{\glsglossarymark}[1]{\glossarymark{#1}}
}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand{\glsglossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUHead{#1}}{\memUHead{#1}}%
      \else
        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand{\glsglossarymark}[1]{%
      \ifglsucmark
        \@mkboth{\glssupercase{#1}}{\glssupercase{#1}}%
      \else
        \@mkboth{#1}{#1}%
      \fi
    }
  }
}
\providecommand{\glossarymark}[1]{%
  \ifglsucmark
    \@mkboth{\glssupercase{#1}}{\glssupercase{#1}}%
  \else
    \@mkboth{#1}{#1}%
  \fi
}
\newcommand*{\setglossarysection}[1]{%
\setkeys{glossaries.sty}{section=#1}}
\newcommand*{\@glossarysection}[2]{%
  \ifdefempty\@glossarysecstar
  {%
    \csname\@glossarysec\endcsname[#1]{#2}%
  }%
  {%
    \csname\@glossarysec\endcsname*{#2}%
    \@gls@toc{#1}{\@glossarysec}%
  }%
  \@glossaryseclabel
}
\newcommand*{\@p@glossarysection}[2]{%
  \glsclearpage
  \phantomsection
  \ifdefempty\@glossarysecstar
  {%

```

```

        \csname\@@glossarysec\endcsname{#2}%
    }%
    {%
        \@gls@toc{#1}\@@glossarysec}%
        \csname\@@glossarysec\endcsname*{#2}%
    }%
    \@@glossaryseclabel
}
\newcommand*\gls@docclearpage{%
\ifthenelse{\equal{\@@glossarysec}{chapter}}%
{%
    \ifcsundef{cleardoublepage}%
    {%
        \clearpage
    }%
    {%
        \ifcsdef{if@openright}%
        {%
            \if@openright
                \cleardoublepage
            \else
                \clearpage
            \fi
        }%
        {%
            \cleardoublepage
        }%
    }%
}%
}%
}%
}
\newcommand*\glscclearpage{\gls@docclearpage}
\newcommand*\@gls@toc}[2]{%
\ifglstoc
\ifglsnumberline
\addcontentsline{toc}{#2}{\protect\numberline{#1}}%
\else
\addcontentsline{toc}{#2}{#1}%
\fi
\fi
}
\newcommand*\glsnoxindywarning}[1]{%
\GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
}
\newcommand*\glsnomakeindexwarning}[1]{%
\GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%
}
\ifglsxindy
\edef\@xdyattributes{\string"default\string"}%
\fi

```



```

\ifglxindy
  \edef\xdyattributelist{}%
\fi
\ifglxindy
  \def\xdylocref{}
\fi
\newcommand*\@gls@ifinlist}[4]{%
  \def\@do@ifinlist##1,#1,##2\end@doifinlist{%
    \def\@gls@listsuffix{##2}%
    \ifx\@gls@listsuffix\@empty
      #4%
    \else
      #3%
    \fi
  }%
  \@do@ifinlist,#2,#1,\end@doifinlist
}
\ifglxindy
  \newcommand*\@xdycounters{\glscounter}
  \newcommand*\GlsAddXdyCounters[1]{%
    \for\@gls@ctr:=#1\do{%
      \edef\@do@addcounter{%
        \noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
        {%
          \noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
            \noexpand\@gls@ctr}%
        }%
      }%
      \do@addcounter
    }
  }
  \onlypremakeg\GlsAddXdyCounters
\else
  \newcommand*\GlsAddXdyCounters[1]{%
    \glsnoindywarning\GlsAddXdyAttribute
  }
\fi
\newcommand*\@disabled@glsaddxdycounters{%
  \PackageError{glossaries}{\string\GlsAddXdyCounters\space
  can't be used after \string\GlsAddXdyAttribute}{Move all
  occurrences of \string\GlsAddXdyCounters\space before the first
  instance of \string\GlsAddXdyAttribute}%
}
\ifglxindy
  \newcommand*\@glsaddxdyattribute[2]{%
    \edef\xdyattributes{\@xdyattributes ^^J
      \string"#2#1\string"}%
    \expandafter\toks@\expandafter{\@xdylocref}%
    \edef\xdylocref{\the\toks@ ^^J%
      (markup-locref

```

```

:open \string"\glstildechar n%
  \expandafter\string\csname glsX#2X#1\endcsname
  \string" ^^J
:close \string"\string" ^^J
:attr \string"#2#1\string"}}%
\expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
  \setentrycounter[##1]{#2}\csname #1\endcsname{##2}%
}%
}
\newcommand*\GlsAddXdyAttribute[1]{%
  \ifx\@xdyattributelist\@empty
  \edef\@xdyattributelist{#1}%
  \else
  \edef\@xdyattributelist{\@xdyattributelist,#1}%
  \fi
  \edef\@xdyattributes{\@xdyattributes ^^J
  \string"#1\string"}%
  \@for\@this@counter:=\@xdycounters\do{%
  \protected@edef\gls@do@addxdyattribute{%
  \noexpand\@glsaddxdyattribute{#1}{\@this@counter}%
  }
  \gls@do@addxdyattribute
  }%
  \let\GlsAddXdyCounters\@disabled@glsaddxdycounters
}
\@onlypremakeg\GlsAddXdyAttribute
\else
\newcommand*\GlsAddXdyAttribute[1]{%
  \glsnoxindywarning\GlsAddXdyAttribute}
\fi
\ifglsexindy
\newcommand*{\@gls@addpredefinedattributes}{%
  \GlsAddXdyAttribute{glsnumberformat}
  \GlsAddXdyAttribute{textrm}
  \GlsAddXdyAttribute{textsf}
  \GlsAddXdyAttribute{texttt}
  \GlsAddXdyAttribute{textbf}
  \GlsAddXdyAttribute{textmd}
  \GlsAddXdyAttribute{textit}
  \GlsAddXdyAttribute{textup}
  \GlsAddXdyAttribute{textsl}
  \GlsAddXdyAttribute{textsc}
  \GlsAddXdyAttribute{emph}
  \GlsAddXdyAttribute{glshypernumber}
  \GlsAddXdyAttribute{hyper rm}
  \GlsAddXdyAttribute{hyper sf}
  \GlsAddXdyAttribute{hyper tt}
  \GlsAddXdyAttribute{hyper bf}
  \GlsAddXdyAttribute{hyper md}
  \GlsAddXdyAttribute{hyper it}
}

```

```

\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}
\GlsAddXdyAttribute{glsignore}
}
\else
\let\@gls@addpredefinedattributes\relax
\fi
\def\@xdyuseralphabets{
\ifglxsindy
\newcommand*\@GlsAddXdyAlphabet}[2]{%
\edef\@xdyuseralphabets{%
\@xdyuseralphabets ^^J
(define-alphabet "#1" (#2))}}
\else
\newcommand*\@GlsAddXdyAlphabet}[2]{%
\glsnoxindywarning\@GlsAddXdyAlphabet}
\fi
\ifglxsindy
\newcommand*\@gls@xdy@locationlist}{%
roman-page-numbers,%
Roman-page-numbers,%
arabic-page-numbers,%
alpha-page-numbers,%
Alpha-page-numbers,%
Appendix-page-numbers,%
arabic-section-numbers%
}
\protected@edef\@gls@roman{\@roman{0}\string"
\string"roman-numbers-lowercase\string" :sep \string"}%
\@onelevel@sanitize\@gls@roman
\edef\@gls@tmp{\string" \string"roman-numbers-lowercase\string"
:sep \string"}%
\@onelevel@sanitize\@gls@tmp
\ifx\@gls@tmp\@gls@roman
\expandafter
\edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
\string"roman-numbers-lowercase\string"%
}%
\else
\expandafter
\edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{
:sep \string"\@gls@roman\string"%
}%
\fi
\expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
\string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%

```

```

    \string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
    \string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
    \string"ALPHA\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
    \string"ALPHA\string"
    :sep \string"\@glsAlphacompositor\string"
    \string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
    \string"arabic-numbers\string"
    :sep \string"\glscompositor\string"
    \string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglsxindy
\newcommand\@xdycrossrefhook{}
\fi
\ifglsxindy
\newcommand*\GlsAddXdyLocation}[3][[]]{%
    \def\@gls@tmp{#1}%
    \ifx\@gls@tmp\@empty
        \edef\@xdyuserlocationdefs{%
            \@xdyuserlocationdefs ^^J%
            (define-location-class \string"#2\string"^^J\space\space
            \space(:sep \string"{}\glsopenbrace\string" #3
            :sep \string"\glsclosebrace\string"))
        }%
    \else
        \edef\@xdyuserlocationdefs{%
            \@xdyuserlocationdefs ^^J%
            (define-location-class \string"#2\string"^^J\space\space
            \space(:sep "\glsopenbrace"
            #1
            :sep "\glsclosebrace\glsopenbrace" #3
            :sep "\glsclosebrace"))
        }%
    \fi
    \edef\@xdyuserlocationnames{%
        \@xdyuserlocationnames^^J\space\space\space
        \string"#2\string"}%
}
\@onlypremakeg\GlsAddXdyLocation
\else

```

```

\newcommand*\GlsAddXdyLocation[2]{%
  \glsnoxywarning\GlsAddXdyLocation}
\fi
\ifglxindy
\def\@xdylocationclassorder{^^J\space\space\space
  \string"roman-page-numbers\string"^^J\space\space\space
  \string"arabic-page-numbers\string"^^J\space\space\space
  \string"arabic-section-numbers\string"^^J\space\space\space
  \string"alpha-page-numbers\string"^^J\space\space\space
  \string"Roman-page-numbers\string"^^J\space\space\space
  \string"Alpha-page-numbers\string"^^J\space\space\space
  \string"Appendix-page-numbers\string"
  \@xdyuserlocationnames^^J\space\space\space
  \string"see\string"
}
\fi
\ifglxindy
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
  \def\@xdylocationclassorder{#1}}
\else
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
  \glsnoxywarning\GlsSetXdyLocationClassOrder}
\fi
\ifglxindy
\def\@xdysortrules{}
\fi
\ifglxindy
\newcommand*\GlsAddSortRule[2]{%
  \expandafter\toks@\expandafter{\@xdysortrules}%
  \protected@edef\@xdysortrules{\the\toks@ ^^J
    (sort-rule \string"#1\string" \string"#2\string")}%
}
\else
\newcommand*\GlsAddSortRule[2]{%
  \glsnoxywarning\GlsAddSortRule}
\fi
\ifglxindy
\def\@xdyrequiredstyles{tex}
\fi
\ifglxindy
\newcommand*\GlsAddXdyStyle[1]{%
  \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
\newcommand*\GlsAddXdyStyle[1]{%
  \glsnoxywarning\GlsAddXdyStyle}
\fi
\ifglxindy
\newcommand*\GlsSetXdyStyles[1]{%
  \edef\@xdyrequiredstyles{#1}}
\else

```

```

\newcommand*\GlsSetXdyStyles[1]{%
  \glsnoxindywarning\GlsSetXdyStyles}
\fi
\def\@xdylanguage#1#2{}
\ifglxsindy
  \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
    \ifglossaryexists{#1}{%
      \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
    }{%
      \PackageError{glossaries}{Can't set language type for
        glossary type '#1' --- no such glossary}{%
        You have specified a glossary type that doesn't exist}}
  \else
    \newcommand*\GlsSetXdyLanguage[2][]{%
      \glsnoxindywarning\GlsSetXdyLanguage}
  \fi
\def\@gls@codepage#1#2{}
\ifglxsindy
  \newcommand*\GlsSetXdyCodePage[1]{%
    \renewcommand*\@gls@codepage{#1}%
  }
  \AtBeginDocument{%
    \ifx\@gls@codepage@empty
      \@ifpackageloaded{fontspec}{\def\@gls@codepage{utf8}}{}%
    \fi
  }
\else
  \newcommand*\GlsSetXdyCodePage[1]{%
    \glsnoxindywarning\GlsSetXdyCodePage}
\fi
\ifglxsindy
  \ifglxsindy@glsnumbers
    \def\@xdylettergroups{(define-letter-group
      \string"glnumbers\string"^^J\space\space\space
      :prefixes (\string"0\string" \string"1\string"
        \string"2\string" \string"3\string" \string"4\string"
        \string"5\string" \string"6\string" \string"7\string"
        \string"8\string" \string"9\string")^^J\space\space\space
      \@xdynumbergrouporder)}
    \else
      \def\@xdylettergroups{}
    \fi
  \fi
\newcommand*\GlsAddLetterGroup[2]{%
  \expandafter\toks@\expandafter{\@xdylettergroups}%
  \protected@edef\@xdylettergroups{\the\toks@^^J%
    (define-letter-group \string"#1\string"^^J\space\space\space#2)}%
  }%
\newcommand*\forallglossaries[3][\@glo@types]{%
  \@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%

```

```

}

\newcommand*\forallacronyms}[2]{%
  \@for#1:=\@glsacronymlists\do{\ifx#1\@empty\else#2\fi}%
}

\newcommand*\forallglsentries}[3][\glsdefaulttype]{%
  \protected@edef\@glo@list{\csname glolist@#1\endcsname}%
  \@for#2:=\@glo@list\do
  {%
    \ifdefempty{#2}{#3}%
  }%
}

\newcommand*\forallglsentries}[3][\@glo@types]{%
  \expandafter\forallglossaries\expandafter[#1]{\@@this@glo@}%
  {%
    \forallglsentries[\@@this@glo@]{#2}{#3}%
  }%
}

\newcommand{\ifglossaryexists}{%
  \@ifstar\s@ifglossaryexists\@ifglossaryexists
}

\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{glo@type@#1@out}{#3}{#2}%
}

\newcommand{\s@ifglossaryexists}[3]{%
  \ifcsundef{glolist@#1}{#3}{#2}%
}

\newcommand*\glsdetoklabel}[1]{#1}
\newcommand{\ifglsentryexists}[3]{%
  \ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}

\newcommand*\ifglsused}[3]{%
  \ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}

\newcommand{\glsdoifexists}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
}

\newcommand{\glsdoifnoexists}[2]{%
  \ifglsentryexists{#1}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
      been defined}{}}{#2}%
}

\newcommand{\glsdoifexistsorwarn}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}%
  }%
}

```

```

}
\newcommand{\glsdoifexistsordo}[3]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}%
    #3%
  }%
}
\newcommand{\doifglossarynoexistsordo}[3]{%
  \ifglossaryexists*{#1}%
  {%
    \PackageError{glossaries}{Glossary type ‘#1’ already exists}{}%
    #3%
  }%
  {#2}%
}
\newrobustcmd{\ifglshaschildren}[3]{%
  \glsdoifexists{#1}%
  {%
    \def\do@glshaschildren{#3}%
    \protected@edef\@gls@thislabel{\glsdetoklabel{#1}}%
    \expandafter\forglseries\expandafter
    [\csname glo@\@gls@thislabel @type\endcsname]
    {\glo@label}%
    {%
      \letcs\glo@parent{glo@\glo@label @parent}%
      \ifdefequal\@gls@thislabel\glo@parent
      {%
        \def\do@glshaschildren{#2}%
        \@endfortrue
      }%
    }%
  }%
  \do@glshaschildren
}
}
\newcommand{\ifglshasparent}[3]{%
  \glsdoifexists{#1}%
  {%
    \ifcsemtyp{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
  }%
}
\newcommand*{\ifglshasdesc}[3]{%
  \ifcsemtyp{glo@\glsdetoklabel{#1}@desc}%
  {#3}%
  {#2}%
}
\newcommand*{\ifglshasdescsuppressed}[3]{%
  \ifcsequal{glo@\glsdetoklabel{#1}@desc}{@no@post@desc}%

```



```

    {#2}%
    {#3}%
  }
\newrobustcmd*{\ifglshassymbol}[3]{%
  \letcs{\@glo@symbol}{glo@glstdetoklabel{#1}@symbol}%
  \ifdefempty\@glo@symbol
    {#3}%
  {%
    \ifdefequal\@glo@symbol\@gls@default@value
      {#3}%
      {#2}%
    }%
  }
}
\newrobustcmd*{\ifglshaslong}[3]{%
  \letcs{\@glo@long}{glo@glstdetoklabel{#1}@long}%
  \ifdefempty\@glo@long
    {#3}%
  {%
    \ifdefequal\@glo@long\@gls@default@value
      {#3}%
      {#2}%
    }%
  }
}
\newrobustcmd*{\ifglshasshort}[3]{%
  \letcs{\@glo@short}{glo@glstdetoklabel{#1}@short}%
  \ifdefempty\@glo@short
    {#3}%
  {%
    \ifdefequal\@glo@short\@gls@default@value
      {#3}%
      {#2}%
    }%
  }
}
\newcommand{\ifglsfieldvoid}[4]{\ifcsvoid{glo@glstdetoklabel{#2}@#1}{#3}{#4}}
\newrobustcmd*{\ifglshasfield}[4]{%
  \glsdoifexists{#2}%
  {%
    \letcs{\@glo@thisvalue}{glo@glstdetoklabel{#2}@#1}%
    \ifdef\@glo@thisvalue
      {%
        \ifdefempty\@glo@thisvalue
          {%
            #4%
          }%
        }%
      }%
    \ifdefequal\@glo@thisvalue\@gls@default@value
      {%
        #4%
      }%
    }%
  }

```

```

        \let\glscurrentfieldvalue\@glo@thisvalue
        #3%
    }%
}%
}%
{%
    \@gls@fetchfield{\@gls@thisfield}{#1}%
    \ifdef\@gls@thisfield
    {%
        \letcs{\@glo@thisvalue}{glo@glstetoklabel{#2}@ \@gls@thisfield}%
        \ifdefempty\@glo@thisvalue
        {%
            #4%
        }%
        {%
            \ifdefequal\@glo@thisvalue\@gls@default@value
            {%
                #4%
            }%
            {%
                \let\glscurrentfieldvalue\@glo@thisvalue
                #3%
            }%
        }%
    }%
    {%
        \GlossariesWarning{Unknown entry field '#1'}%
        #4%
    }%
}%
}%
}
\newcommand*{\glscurrentfieldvalue}{}
\newcommand*{\@glo@types}{,}
\newcommand*{\@gls@provide@newglossary}{%
    \protected@write\@auxout{}{\string\providecommand\string\@newglossary[4]{}}%
    \let\@gls@provide@newglossary\relax
}
\newcommand*{\@gls@defaulttype}{\glsdefaulttype}
\newcommand*{\defglsentryfmt}[2][\glsdefaulttype]{%
    \protected@edef\@glo@tmp{#1}%
    \ifx\@glo@tmp\@gls@defaulttype
        \GlossariesWarning{\string\defglsentryfmt\space attempting to set
            a display style for \string\glsdefaulttype\space but there
            doesn't seem to be a default glossary. If you have used
            'nomain', you need to define an alternative glossary}%
    \else
        \csgdef{gls@\@glo@tmp @entryfmt}{#2}%
    \fi
}
}

```

```

\newcommand*\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*\@gls@forbidtext}[1]{%
\ifboolexpr{test {\ifdefstring{#1}{tex}}
or test {\ifdefstring{#1}{TEX}}}
{%
\def#1{nottex}%
\PackageError{glossaries}%
{Forbidden '.tex' extension replaced with '.nottex'}%
{I'm sorry, I can't allow you to do something so reckless.\MessageBreak
Don't use '.tex' as an extension for a temporary file.}%
}%
}%
}%
}
\newcommand*\gls@gobbleopt}{\new@ifnextchar[{\@gls@gobbleopt}{}]}
\def\@gls@gobbleopt[#1]{%
\newcommand*\@newglossary}{\ifstar\s@newglossary\ns@newglossary}
\newcommand*\@s@newglossary}[2]{%
\ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
}
\newcommand*\@ns@newglossary}[5][glg]{%
\doifglossarynoexistsordo{#2}%
{%
\ifundef\glsdefaulttype
{%
\gdef\glsdefaulttype{#2}%
}{}%
\toks@{#2}\protected@edef\@glo@types{\@glo@types\the\toks@,}%
\expandafter\gdef\csname glolist@#2\endcsname{,}%
\expandafter\edef\csname @glotype@#2@log\endcsname{#1}%
\expandafter\edef\csname @glotype@#2@in\endcsname{#3}%
\expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
\expandafter\@gls@forbidtext\csname @glotype@#2@log\endcsname
\expandafter\@gls@forbidtext\csname @glotype@#2@in\endcsname
\expandafter\@gls@forbidtext\csname @glotype@#2@out\endcsname
\expandafter\def\csname @glotype@#2@title\endcsname{#5}%
\@gls@provide@newglossary
\protected@write\@auxout}{\string\@newglossary{#2}{#1}{#3}{#4}}%
\ifcsundef{gls@#2@entryfmt}%
{%
\defglsentryfmt[#2]{\glsentryfmt}%
}%
}%
\@gls@defsortcount{#2}%
\@ifnextchar[{\@gls@setcounter{#2}}%
{\@gls@setcounter{#2}[\glscounter]}%
}%
}%
\gls@gobbleopt
}%

```

```

}
\newcommand*\altnewglossary}[3]{%
  \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
}
\@onlypreamble{\newglossary}
\@onlypremakeg\newglossary
\newcommand*\@newglossary}[4]{%
\def\@gls@setcounter#1[#2]{%
  \expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
  \ifglsxindy
    \GlsAddXdyCounters{#2}%
  \fi
}
\newcommand*\@gls@getcounter}[1]{%
  \csname @glotype@#1@counter\endcsname
}
\glsdefmain
\@gls@do@acronymsdef
\@gls@do@symbolsdef
\@gls@do@numbersdef
\@gls@do@indexdef
\newcommand*\newignoredglossary}[1]{%
  \ifdefempty\@ignored@glossaries
  {%
    \protected@edef\@ignored@glossaries{#1}%
  }%
  {%
    \protected@eappto\@ignored@glossaries{, #1}%
  }%
  \csgdef{glolist@#1}{,}%
  \ifcsundef{gls@#1@entryfmt}%
  {%
    \defglsentryfmt[#1]{\glsentryfmt}%
  }%
  {}%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*\@gls@nohyperlist}{#1}%
  }%
  {%
    \protected@eappto\@gls@nohyperlist{, #1}%
  }%
}
\newcommand*\@ignored@glossaries{}
\newcommand*\ifignoredglossary}[3]{%
  \protected@edef\@gls@igtype{#1}%
  \expandafter\DTLifinlist\expandafter
  {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
}
\define@key{glossentry}{name}{%

```

```

\def\@glo@name{#1}%
}
\define@key{glossentry}{description}{%
\def\@glo@desc{#1}%
}
\define@key{glossentry}{descriptionplural}{%
\def\@glo@descplural{#1}%
}
\define@key{glossentry}{sort}{%
\def\@glo@sort{#1}}
\define@key{glossentry}{text}{%
\def\@glo@text{#1}%
}
\define@key{glossentry}{plural}{%
\def\@glo@plural{#1}%
}
\define@key{glossentry}{first}{%
\def\@glo@first{#1}%
}
\define@key{glossentry}{firstplural}{%
\def\@glo@firstplural{#1}%
}
\newcommand*\@gls@default@value{\relax}
\define@key{glossentry}{symbol}{%
\def\@glo@symbol{#1}%
}
\define@key{glossentry}{symbolplural}{%
\def\@glo@symbolplural{#1}%
}
\define@key{glossentry}{type}{%
\def\@glo@type{#1}}
\define@key{glossentry}{counter}{%
\ifcsundef{c@#1}%
{%
\PackageError{glossaries}
{There is no counter called ‘#1’}%
{%
The counter key should have the name of a valid counter
as its value%
}%
}%
\def\@glo@counter{#1}%
}%
}
\define@key{glossentry}{see}{%
\gls@set@xr@key{see}{\@glo@see}{#1}%
}
\newcommand*\@gls@set@xr@key[3]{%
\renewcommand*\@gls@xr@key{#1}%
}

```

```

\gls@checkseeallowed
\def#2{#3}%
\@glo@seeautonumberlist
}
\newcommand*{\gls@xr@key}{see}
\newcommand*{\gls@checkseeallowed}{%
\@gls@see@noindex
}
\newcommand*{\gls@checkseeallowed@preambleonly}{%
\GlossariesWarning
{'\gls@xr@key' key doesn't have any effect when used in the document
environment. Move the definition to the preamble
after \string\makeglossaries\space
or \string\makenoidxglossaries}%
}
\define@key{glossentry}{parent}{%
\def\@glo@parent{#1}}
\define@choicekey{glossentry}{nonumberlist}{%
[\@gls@nonumberlist@val\@gls@nonumberlist@nr]{true,false}[true]}%
{
\ifcase\@gls@nonumberlist@nr\relax
\def\@glo@prefix{\@gls@nonextpages}%
\@gls@savenonumberlist{true}%
\else
\def\@glo@prefix{\@gls@nextpages}%
\@gls@savenonumberlist{false}%
\fi
}
\newcommand*{\@gls@noidx@nonumberlist}[1]{%
\cslet\@glo@\@glo@label @prenumberlist}\@glo@prefix
}%
\newcommand*\@gls@noidx@prenumberlist}[1]{%
\csuse\@glo@\@gls@detoklabel{#1}@prenumberlist}
\newcommand*{\@gls@savenonumberlist}[1]{}
\newcommand*{\@gls@initnonumberlist}{}%
\newcommand*{\@gls@storenonumberlist}[1]{}
\newcommand*{\@gls@enablesavenonumberlist}{%
\renewcommand*{\@gls@initnonumberlist}{%
\undef\@glo@nonumberlist
}%
\renewcommand*{\@gls@savenonumberlist}[1]{%
\def\@glo@nonumberlist{##1}%
}%
\renewcommand*{\@gls@storenonumberlist}[1]{%
\ifdef\@glo@nonumberlist
{
\cslet\@glo@\@gls@detoklabel{##1}@nonumberlist}\@glo@nonumberlist}%
}%
{}%
}%

```

```

\appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}
\define@key{glossentry}{user1}{%
  \def\@glo@useri{#1}%
}
\define@key{glossentry}{user2}{%
  \def\@glo@userii{#1}%
}
\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#1}%
}
\define@key{glossentry}{short}{%
  \def\@glo@short{#1}%
}
\define@key{glossentry}{shortplural}{%
  \def\@glo@shortpl{#1}%
}
\define@key{glossentry}{long}{%
  \def\@glo@long{#1}%
}
\define@key{glossentry}{longplural}{%
  \def\@glo@longpl{#1}%
}
\newcommand*\@glsnoname}{%
  \PackageError{glossaries}{name key required in
  \string\newglossaryentry\space for entry '@glo@label'}{You
  haven't specified the entry name}}
\newcommand*\@glsnodesc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}%
\newcommand*\@gls@missingnumberlist}[1]{%
  ??%
  \ifglsavenumberlist
  \GlossariesWarning{Missing number list for entry '#1'}.

```

```

    Maybe makeglossaries + rerun required}%
\else
\PackageError{glossaries}%
{Package option 'savenumberlist=true' required}%
{%
    You must use the 'savenumberlist' package option
    to reference location lists.%
}%
\fi
}
\newcommand*{\@glsdefaultsort}{\@glo@name}
\newcount\gls@level
\newcommand{\@gls@noexpand@field}[3]{%
\expandafter\global\expandafter
\let\csname glo@#1@#2\endcsname#3%
}
\newcommand{\@gls@noexpand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\csuse{gls@assign@#3@field}{#2}{#4}%
}%
}%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\@gls@noexpand@field{#2}{#3}{\@gls@value}%
}%
{%
\@gls@noexpand@field{#2}{#3}{#4}%
}%
}%
}
\newcommand{\@gls@expand@field}[3]{%
\expandafter
\protected@xdef\csname glo@#1@#2\endcsname{#3}%
}
\newcommand{\@gls@expand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}
}
}
}

```



```

}%
{%
  \expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
  {%
    \@@gls@expand@field{#2}{#3}{#4}%
  }%
  {%
    \csuse{gls@assign@#3@field}{#2}{#4}%
  }%
}%
}%
{%
  \ifdefequal{#4}{\@gls@default@value}%
  {%
    \@@gls@expand@field{#2}{#3}{#1}%
  }%
  {%
    \@@gls@expand@field{#2}{#3}{#4}%
  }%
}%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
  \def\@gls@tmp{#1}%
  \ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*\gls@expandfields{%
  \let\gls@assign@field\@gls@expand@fields
}
\newcommand*\gls@noexpandfields{%
  \let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
  \glsdoifnoexists{#1}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\newcommand*\gls@defdocnewglossaryentry{%
  \let\newglossaryentry\new@glossaryentry
}
\newrobustcmd{\provideglossaryentry}[2]{%
  \ifglsentryexists{#1}%
  {}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\@onlypreamble{\provideglossaryentry}

```

```

\newcommand*\gls@glsdefs@labellist}{
\newcommand*\gls@glsdefs@addlabel}[1]{\listxadd{\gls@glsdefs@labellist}{#1}}
\newcommand*\gls@docdef@labellist}{
\newcommand*\gls@docdef@addlabel}[1]{\listxadd{\gls@docdef@labellist}{#1}}
\newrobustcmd\new@glossaryentry}[2]{%
  \ifundef\@gls@deffile
  {%
    \global\newwrite\@gls@deffile
    \immediate\openout\@gls@deffile=\jobname.glsdefs
  }%
  }%
\ifglsentryexists{#1}%
{%
  \xifinlist{#1}{\gls@docdef@labellist}%
  {%
    \GlossariesWarning{Entry ‘#1’ has already been defined in the
      document environment (ignoring new definition)}%
  }%
  {%
    \xifinlist{#1}{\gls@glsdefs@labellist}%
    {%
      \gls@defglossaryentry{#1}{#2}%
      \gls@docdef@addlabel{#1}%
      \@gls@writedef{#1}%
    }%
    {%
      \GlossariesWarning{Entry ‘#1’ has already been defined in
        the preamble (in which case remove the duplicate definition)
        or you have just updated to glossaries.sty
        v4.47 (in which case delete ‘\jobname.glsdefs’)}%
    }%
  }%
}
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*\gls@begindocdefs){%
  \@gls@enablesavenonumberlist
  \edef\@gls@restreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
  \makeatletter
  \InputIfFileExists{\jobname.glsdefs}{}{}%
  \@gls@restreat
  \undef\@gls@restreat
  \gls@defdocnewglossaryentry
}

```

```

\newcommand*\glsdefs@newdocentry}[2]{%
  \ifglsentryexists{#1}%
    {\GlossariesWarning{entry ‘#1’ has already been defined in the preamble
      (if you have moved the definition there from the document
        environment, delete \jobname.glsdefs)}}%
    {%
      \gls@glsdefs@addlabel{#1}%
      \gls@defglossaryentry{#1}{#2}%
    }%
}
\newcommand*\@gls@writedef}[1]{%
  \immediate\write\@gls@deffile
  {%
    \string\glsdefs@newdocentry{\glsdetoklabel{#1}}\glspercentchar^^J%
    \expandafter@gobble\string\{\glspercentchar%
  }%
  \@for\@gls@map:=\@gls@keymap\do
  {%
    \letcs@glo@value{glo@\glsdetoklabel{#1}@\expandafter\@secondoftwo\@gls@map}%
    \ifdef@glo@value
      {%
        \@onelevel@sanitize@glo@value
        \immediate\write\@gls@deffile
        {%
          \expandafter\@firstoftwo\@gls@map
          =\expandafter@gobble\string\{\glo@value\expandafter@gobble\string\},%
          \glspercentchar
        }%
      }%
    }%
  }%
  \gls@writedefhook
  \immediate\write\@gls@deffile
  {%
    \glspercentchar^^J%
    \expandafter@gobble\string\}\glspercentchar^^J%
  }%
}
\newcommand*\@gls@keymap}{%
  {name}{name},%
  {sort}{sortvalue},% unescaped sort value
  {type}{type},%
  {first}{first},%
  {firstplural}{firstpl},%
  {text}{text},%
  {plural}{plural},%
  {description}{desc},%
  {descriptionplural}{descplural},%
  {symbol}{symbol},%
  {symbolplural}{symbolplural},%

```

```

{user1}{useri},%
{user2}{userii},%
{user3}{useriii},%
{user4}{useriv},%
{user5}{userv},%
{user6}{uservi},%
{long}{long},%
{longplural}{longpl},%
{short}{short},%
{shortplural}{shortpl},%
{counter}{counter},%
{parent}{parent}%
}
\newcommand*{\@gls@fetchfield}[2]{%
  \edef\@gls@thisval{#2}%
  \@for\@gls@map:=\@gls@keymap\do{%
    \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
    \ifdefequal{\@this@key}{\@gls@thisval}%
      {%
        \edef#1{\expandafter\@secondoftwo\@gls@map}%
        \@endfortrue
      }%
    }%
  }%
}
\newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
\newcommand*{\@sglsaddstoragekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
      {\csname gls@assign@#1@field\endcsname}[2]{%
        \@gls@expand@field{##1}{#1}{##2}%
      }%
  }%
  }%
  }%
  \@glsaddstoragekey{#1}%
}
\newcommand*{\@glsaddstoragekey}[3]{-%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{, {#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{\@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
  }%
  \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
}
}

```

```

    \PackageError{glossaries}{Key ‘#1’ already exists}{}%
  }%
}
\newcommand*\glsaddkey{\@ifstar\@sglsaddkey\@glsaddkey}
\newcommand*\@sglsaddkey[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
      {\csname gls@assign@#1@field\endcsname}[2]{%
        \@gls@expand@field{##1}{#1}{##2}%
      }%
  }%
}%
\@glsaddkey{#1}%
}
\newcommand*\@glsaddkey[7]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{, {#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
    \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
    \newcommand*{#4}[1]{\@Glsentryfield{##1}{#1}}%
    \ifcsdef{@gls@user@#1@}%
    {%
      \PackageError{glossaries}%
      {Can’t define ‘\string#5’ as helper command
        ‘\expandafter\string\csname @gls@user@#1@\endcsname’ already exists}%
    }%
  }%
  {%
    \expandafter\newcommand\expandafter*\expandafter
      {\csname @gls@user@#1\endcsname}[2] [] {%
        \new@ifnextchar [%
          {\csuse{@gls@user@#1@}{##1}{##2}}%
          {\csuse{@gls@user@#1@}{##1}{##2} []}}%
        \csdef{@gls@user@#1@}##1##2[##3]{%
          \@gls@field@link{##1}{##2}{#3{##2}##3}%
        }%
      }%
    \newrobustcmd*{#5}{%
      \expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
    }%
    \ifcsdef{@Gls@user@#1@}%
    {%
      \PackageError{glossaries}%
      {Can’t define ‘\string#6’ as helper command

```

```

        '\expandafter\string\csname @Gls@user@#1@\endcsname' already exists}%
    }%
}%
{
\expandafter\newcommand\expandafter*\expandafter
  {\csname @Gls@user@#1@\endcsname}[2] [] {%
    \new@ifnextchar [%
      {\csuse{@Gls@user@#1@}{##1}{##2}}%
      {\csuse{@Gls@user@#1@}{##1}{##2} []}}%
\csdef{@Gls@user@#1@}##1##2[##3] {%
  \gls@field@link{##1}{##2}{#4{##2}##3}%
}%
\newrobustcmd*{#6}{%
  \expandafter\@gls@hyp@opt\csname @Gls@user@#1@\endcsname}%
}%
\ifcsdef{@GLS@user@#1@}%
{
  \PackageError{glossaries}%
  {Can't define '\string#7' as helper command
  '\expandafter\string\csname @GLS@user@#1@\endcsname' already exists}%
  {}%
}%
{
\expandafter\newcommand\expandafter*\expandafter
  {\csname @GLS@user@#1@\endcsname}[2] [] {%
    \new@ifnextchar [%
      {\csuse{@GLS@user@#1@}{##1}{##2}}%
      {\csuse{@GLS@user@#1@}{##1}{##2} []}}%
\csdef{@GLS@user@#1@}##1##2[##3] {%
  \@gls@field@link{##1}{##2}{\glsuppercase{#3{##2}##3}}%
}%
\newrobustcmd*{#7}{%
  \expandafter\@gls@hyp@opt\csname @GLS@user@#1@\endcsname}%
}%
\glsmfuaddmap{#3}{#4}%
\glsmfuaddmap{#5}{#6}%
\glsmfublocker{#7}%
}%
{
  \PackageError{glossaries}{Key '#1' already exists}{}%
}%
}
\newrobustcmd{\glsfieldxdef}[3] {%
\glsdoifexists{#1}%
{
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {
    \protected@csxdef{glo@\@glo@label @#2}{#3}%
  }%
}
}

```

```

    {%
      \PackageError{glossaries}{Field '#2' doesn't exist}{}%
    }%
  }%
}
\newrobustcmd{\glsfielddedf}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \protected@csedef{glo@\@glo@label @#2}{#3}%
    }%
    {%
      \PackageError{glossaries}{Field '#2' doesn't exist}{}%
    }%
  }%
}
\newrobustcmd{\glsfieldgdef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
    }%
    {%
      \PackageError{glossaries}{Field '#2' doesn't exist}{}%
    }%
  }%
}
\newrobustcmd{\glsfieldddf}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
    }%
    {%
      \PackageError{glossaries}{Field '#2' doesn't exist}{}%
    }%
  }%
}
\newrobustcmd{\glsfieldfetch}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%

```

```

        \letcs#3{glo@\@glo@label @#2}%
    }%
    {%
        \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
    }%
}
}
\newcommand*{\glsunexpandedfieldvalue}[2]{%
    \ifcsname glo@\glsdetoklabel{#1}@#2\endcsname
        \expandafter\expandonce\csname glo@\glsdetoklabel{#1}@#2\expandafter\endcsname
    \fi
}
\newrobustcmd{\ifglsfieldeq}[5]{%
    \glsdoifexists{#1}%
    {%
        \protected@edef\@glo@label{\glsdetoklabel{#1}}%
        \ifcsdef{glo@\@glo@label @#2}%
        {%
            \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
        }%
        {%
            \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
        }%
    }%
}
\newrobustcmd{\ifglsfielddefeq}[5]{%
    \glsdoifexists{#1}%
    {%
        \protected@edef\@glo@label{\glsdetoklabel{#1}}%
        \ifcsdef{glo@\@glo@label @#2}%
        {%
            \expandafter\ifdefstrequal\csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
        }%
        {%
            \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
        }%
    }%
}
\newcommand{\ifglsfieldcseq}[5]{%
    \glsdoifexists{#1}%
    {%
        \protected@edef\@glo@label{\glsdetoklabel{#1}}%
        \ifcsdef{glo@\@glo@label @#2}%
        {%
            \ifcsstrequal{glo@\@glo@label @#2}{#3}{#4}{#5}%
        }%
        {%
            \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
        }%
    }%
}

```



```

}
\newcommand*\glswritedefhook{}
\newcommand*\gls@assign@desc}[1]{%
  \gls@assign@field{#1}{desc}{\@glo@desc}%
  \gls@assign@field{\@glo@desc}{#1}{descplural}{\@glo@descplural}%
}
\newcommand{\longnewglossaryentry}[3]{%
  \glsdoifnoexists{#1}%
  {%
    \bgroup
    \let\@org@newglossaryentryprehook\@newglossaryentryprehook
    \long\def\@newglossaryentryprehook{%
      \long\def\@glo@desc{#3\leavevmode\unskip\nopostdesc}%
      \@org@newglossaryentryprehook
    }%
    \renewcommand*\gls@assign@desc}[1]{%
      \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
      \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@desc}%
    }
    \gls@defglossaryentry{#1}{#2}%
  \egroup
}
}
\@onlypreamble{\longnewglossaryentry}
\newcommand{\longprovideglossaryentry}[3]{%
  \ifglsentryexists{#1}{%
    {\longnewglossaryentry{#1}{#2}{#3}}%
  }
}
\@onlypreamble{\longprovideglossaryentry}
\newcommand{\gls@defglossaryentry}[2]{%
  \let\GlsSetQuote\gls@nosetquote
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \let\glslabel\@glo@label
  \let\@glo@name\@glsnoname
  \let\@glo@desc\@glsnodesc
  \let\@glo@descplural\@gls@default@value
  \let\@glo@type\@gls@default@value
  \let\@glo@symbol\@gls@default@value
  \let\@glo@symbolplural\@gls@default@value
  \let\@glo@text\@gls@default@value
  \let\@glo@plural\@gls@default@value
  \let\@glo@first\@gls@default@value
  \let\@glo@firstplural\@gls@default@value
  \let\@glo@sort\@gls@default@value
  \let\@glo@counter\@gls@default@value
  \def\@glo@see{}%
  \def\@glo@parent{}%
  \def\@glo@prefix{}%
  \@gls@initnonumberlist
  \def\@glo@useri{}%

```

```

\def@glo@userii{%
\def@glo@useriii{%
\def@glo@useriv{%
\def@glo@userv{%
\def@glo@uservi{%
\def@glo@short{%
\def@glo@shortpl{%
\def@glo@long{%
\def@glo@longpl{%
\@newglossaryentryprehook
\setkeys{glossentry}{#2}%
\ifundef\glsdefaulttype
{%
  \PackageError{glossaries}%
  {No default glossary type (have you used ‘nomain’ by mistake?)}%
  {If you use package option ‘nomain’ you must define
   a new glossary before you can define entries}%
}%
{}%
\gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
\protected@edef@glo@type{\glsentrytype{\@glo@label}}%
\ifcsundef{glolist@\@glo@type}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘\@glo@type’ has not been defined}%
  {You need to define a new glossary type, before making entries
   in it}%
}%
{%
  \ifignoredglossary@glo@type
  {%
    \ifx@glo@desc@glsnodesc
    \let@glo@desc@empty
    \fi
  }%
  {%
  }%
  \protected@edef@glolist@{\csname glolist@\@glo@type\endcsname}%
  \expandafter\protected@xdef\csname glolist@\@glo@type\endcsname{%
    \@glolist@{\@glo@label},}%
  }%
\gls@level=0\relax
\ifx@glo@parent@empty
\expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
\else
\ifdequal@glo@label@glo@parent%
{%
  \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
  \def@glo@parent{}%
  \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%

```

```

}%
{%
\ifglsentryexists{\@glo@parent}%
{%
\expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
\@glo@parent}%
\gls@level=\csname glo@\@glo@parent @level\endcsname\relax
\advance\gls@level by 1\relax
\ifx\@glo@name\@glsnoname
\expandafter\let\expandafter\@glo@name
\csname glo@\@glo@parent @name\endcsname
\ifx\@glo@plural\@gls@default@value
\expandafter\let\expandafter\@glo@plural
\csname glo@\@glo@parent @plural\endcsname
\fi
\fi
}%
{%
\PackageError{glossaries}%
{%
Invalid parent ‘\@glo@parent’
for entry ‘\@glo@label’ - parent doesn’t exist%
}%
{%
Parent entries must be defined before their children%
}%
\def\@glo@parent{}%
\expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
}%
}%
\fi
\expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%
\gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
\letcs\@glo@sort{glo@\@glo@label @sortvalue}%
\gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @text\endcsname\glspluralsuffix}%
{\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @text\endcsname}%
{\@glo@label}{first}{\@glo@first}%
\ifx\@glo@first\@gls@default@value
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @plural\endcsname}%
{\@glo@label}{firstpl}{\@glo@firstplural}%
\else
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
{\@glo@label}{firstpl}{\@glo@firstplural}%
\fi

```

```

\ifcsundef{@glo@type@\glo@type @counter}%
{%
  \def\@glo@defaultcounter{\glscounter}%
}%
{%
  \letcs\@glo@defaultcounter{@glo@type@\glo@type @counter}%
}%
\gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
\gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
\gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
\gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
\gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
\gls@assign@field{}{\@glo@label}{userv}{\@glo@userv}%
\gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
\gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
\gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
\gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
\gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%
\ifx\@glo@name\@glsnname
  \glsnname
  \let\@glo@name\@gls@default@value
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%
\ifcsundef{glo@\@glo@label @numberlist}%
{%
  \csxdef{glo@\@glo@label @numberlist}{%
    \noexpand\@gls@missingnumberlist{\@glo@label}}%
}%
}%
\@gls@storenonumberlist{\@glo@label}%
\def\@glo@@desc{\@glo@first}%
\ifx\@glo@desc\@glo@@desc
  \let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@glsnodesc
  \glsnodesc
  \let\@glo@desc\@gls@default@value
\fi
\gls@assign@desc{\@glo@label}%
\@gls@defsort{\@glo@type}{\@glo@label}%
\def\@glo@@symbol{\@glo@text}%
\ifx\@glo@symbol\@glo@@symbol
  \let\@glo@symbol\@glo@text
\fi
\gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
\expandafter
  \gls@assign@field\expandafter
    {csname glo@\@glo@label @symbol\endcsname}
    {\@glo@label}{symbolplural}{\@glo@symbolplural}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagfalse\endcsname{%

```

```

\noexpand\global
\noexpand\let\expandafter\noexpand
\csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagtrue\endcsname{%
\noexpand\global
\noexpand\let\expandafter\noexpand
\csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
}%
\csname glo@\@glo@label @flagfalse\endcsname
\@glo@autosee
\ifignoredglossary\@glo@type
{%
\csdef{glo@\@glo@label @index}{}%
}
{%
\do@glo@storeentry{\@glo@label}%
}%
\@newglossaryentry@defcounters
\@newglossaryentryposthook
}
\newcommand*{\@glo@autosee}{%
\ifdefvoid\@glo@see{%
{%
\protected@edef\@do@glssee{%
\noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
\noexpand\expandafter\noexpand\@glssee\noexpand\@glo@list{\@glo@label}}%
\@do@glssee
}%
\@glo@autoseehook
}%
\newcommand*{\@glo@autoseehook}{%
\newcommand*{\@newglossaryentryprehook}{%
\newcommand*{\@newglossaryentryposthook}{%
\newcommand*{\@newglossaryentry@defcounters}{%
\newcommand*{\glsmoveentry}[2]{%
\protected@edef\@glo@thislabel{\glsdetoklabel{#1}}%
\protected@edef\glo@type{\csname glo@\@glo@thislabel @type\endcsname}%
\def\glo@list{,%
\for@gl@sentries[\@glo@type]{\@glo@label}%
{%
\ifdefequal\@glo@thislabel\glo@label
}{\protected@eappto\glo@list{\@glo@label,}}%
}%
\cslet\glo@list@\@glo@type{\@glo@list}%
\csdef{glo@\@glo@thislabel @type}{#2}%
}
\ifglxsindy
\newcommand*{\@glossaryentryfield}{\string\@glossentry}
\else

```

```

\newcommand*{\@glossaryentryfield}{\string\glossentry}
\fi
\ifglxindy
\newcommand*{\@glossarysubentryfield}{%
\string\subglossentry}
\else
\newcommand*{\@glossarysubentryfield}{%
\string\subglossentry}
\fi
\newcommand{\@glo@storeentry}[1]{%
\protected@edef\@glo@esclabel{#1}%
\@gls@checkmkidxchars\@glo@esclabel
\protected@edef\@glo@sort{\csname glo@#1@sort\endcsname}%
\@gls@checkmkidxchars\@glo@sort
\@gls@checkmkidxchars\@glo@prefix
\protected@edef\@glo@parent{\csname glo@#1@parent\endcsname}%
\ifglxindy
\ifx\@glo@parent\@empty
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
(\string"\@glo@sort\string" %
\string"\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %
}%
\else
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
\csname glo@\@glo@parent @index\endcsname
(\string"\@glo@sort\string" %
\string"\@glo@prefix\@glossarysubentryfield
{\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
}%
\fi
\else
\ifx\@glo@parent\@empty
\@onelevel@sanitize\@glo@prefix
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
\@glo@sort\@gls@actualchar\@glo@prefix
\@glossaryentryfield{\@glo@esclabel}%
}%
\else
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
\csname glo@\@glo@parent @index\endcsname\@gls@levelchar
\@glo@sort\@gls@actualchar\@glo@prefix
\@glossarysubentryfield
{\csname glo@#1@level\endcsname}{\@glo@esclabel}%
}%
\fi
\fi
}
\newcommand*{\@gls@if@measuring}[2]{%
\ifmeasuring@
#1%

```

```

\else
  #2%
\fi
}
\let\gls@org@refstepcounter\refstepcounter
\let\gls@org@label\label
\newcommand{\gls@if@measuring}[2]{#2}
\AtBeginDocument{%
  \ifpackageloaded{amsmath}%
  {\let\gls@if@measuring\@gls@if@measuring}%
  }%
  \let\gls@org@refstepcounter\refstepcounter
  \let\gls@org@label\label
}
\newcommand*\gls@ifnotmeasuring[1]{\glsifmeasuring{#1}}
\newcommand*\glsifmeasuring{\gls@if@measuring}
\newcommand*\gls@local@stepcounter}[1]{%
  \ifcsdef{c#1}%
  {%
    \epreto\gls@measuring@reset{\noexpand\setcounter{#1}{\number\value{#1}}}%
    \expandafter\advance\csname c@#1\endcsname by \@ne
  }%
  {}%
}
}
\newcommand{\gls@start@measuring}{%
  \ifx\glsifmeasuring\@firstoftwo
  \else
    \let\gls@org@target\glstarget
    \let\gls@org@link\@glslink
    \def\gls@measuring@reset{}%
  \fi
  \let\glsifmeasuring\@firstoftwo
  \let\refstepcounter\gls@local@stepcounter
  \let\label\@gobble
  \let\glstarget\@secondoftwo
  \let\@glslink\@secondoftwo
}
\ExplSyntaxOn
\tl_if_exist:NT \l_datatool_measure_hook_tl
{
  \tl_put_right:Nn \l_datatool_measure_hook_tl
  { \gls@start@measuring }
}
\ExplSyntaxOff
\newcommand{\gls@stop@measuring}{%
  \renewcommand*\glsifmeasuring{\gls@if@measuring}%
  \let\refstepcounter\gls@org@refstepcounter
  \let\label\gls@org@label
  \let\glstarget\gls@org@target
  \let\@glslink\gls@org@link
}

```

```

\gls@measuring@reset
\def\gls@measuring@reset{%
}
\newcommand{\glsmeasureheight}[2]{%
\settoheight{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\newcommand{\glsmeasuredepth}[2]{%
\settodepth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\newcommand{\glsmeasurewidth}[2]{%
\settowidth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\def\@gls@patchtabularx#1\hbox#2#3!!{%
\def\TX@trial##1{#1\hbox{\gls@start@measuring#2\gls@stop@measuring}#3}%
}
\newcommand*\glspatchtabularx{%
\ifdef\TX@trial
{%
\expandafter\@gls@patchtabularx\TX@trial{##1}!!%
\let\glspatchtabularx\relax
}%
}%
}
\newcommand*\glsreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glsreset{#1}%
}%
}%
}
\newcommand*\glslocalreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glslocalreset{#1}%
}%
}%
}
\newcommand*\glsunset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glsunset{#1}%
}%
}%
}
}

```



```

\newcommand*\glslocalunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glslocalunset{#1}%
    }%
  }%
}
\newcommand*\@glslocalunset{\@glslocalunset}
\newcommand*\@glslocalunset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
}
\newcommand*\@glsunset{\@glsunset}
\newcommand*\@glsunset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
}
\newcommand*\@glslocalreset{\@glslocalreset}
\newcommand*\@glslocalreset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
}
\newcommand*\@glsreset{\@glsreset}
\newcommand*\@glsreset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
}
\newcommand*\glsresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsreset{\@glsentry}%
  }%
}
\newcommand*\glslocalresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalreset{\@glsentry}%
  }%
}
\newcommand*\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsunset{\@glsentry}%
  }%
}
\newcommand*\glslocalunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalunset{\@glsentry}%
  }%
}
\newcommand*\@newglossaryentry@defcounters{%

```

```

\csdef{glo@\glo@label @currcount}{0}%
\csdef{glo@\glo@label @prevcount}{0}%
}
\newif\ifglsresetcurrcount
\glsresetcurrcountfalse
\newcommand*\glsenableentrycount}{%
\let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
\renewcommand*\gls@defdocnewglossaryentry}{%
\renewcommand*newglossaryentry[2]{%
\PackageError{glossaries}{\string\newglossaryentry\space
may only be used in the preamble when entry counting has
been activated}{If you use \string\glsenableentrycount\space
you must place all entry definitions in the preamble not in
the document environment}%
}%
}%
\newcommand*\glsentrycurrcount}[1]{%
\ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
{0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*\glsentryprevcount}[1]{%
\ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
{0}{\@gls@entry@field{##1}{prevcount}}%
}%
\renewcommand*\@glsunset}[1]{%
\@glsunset{##1}%
\@gls@increment@currcount{##1}%
}%
\renewcommand*\@glslocalunset}[1]{%
\@glslocalunset{##1}%
\@gls@local@increment@currcount{##1}%
}%
\renewcommand*\@glsreset}[1]{%
\@glsreset{##1}%
\ifglsresetcurrcount\csgdef{glo@\glsdetoklabel{##1}@currcount}{0}\fi
}%
\renewcommand*\@glslocalreset}[1]{%
\@glslocalreset{##1}%
\ifglsresetcurrcount\csdef{glo@\glsdetoklabel{##1}@currcount}{0}\fi
}%
\def\@cgl@##1##2[##3]{%
\ifnum\glsentryprevcount{##2}=1\relax
\cglformat{##2}{##3}%
\glsunset{##2}%
\else
\@gls@{##1}{##2}[##3]%
\fi
}%
\def\@cgl@##1##2[##3]{%
\ifnum\glsentryprevcount{##2}=1\relax

```

```

        \cglspformat{##2}{##3}%
        \glset{##2}%
    \else
        \cglsp@{##1}{##2}[##3]%
    \fi
}%
\def\cGls@##1##2[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \cGls@{##1}{##2}[##3]%
    \fi
}%
\def\cGlspl@##1##2[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglspformat{##2}{##3}%
        \glset{##2}%
    \else
        \cGlspl@{##1}{##2}[##3]%
    \fi
}%
\AtEndDocument{\@gls@write@entrycounts}%
\renewcommand*{\@gls@entry@count}[2]{%
    \csdef{glo@glsdetoklabel{##1}@prevcount}{##2}%
}%
\let\glsetenableentrycount\relax
}
\@onlypreamble\glsetenableentrycount
\newcommand*{\@gls@increment@currcount}[1]{%
    \csdef{glo@glsdetoklabel{##1}@currcount}{%
        \number\numexpr\glsetentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@local@increment@currcount}[1]{%
    \csdef{glo@glsdetoklabel{##1}@currcount}{%
        \number\numexpr\glsetentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@write@entrycounts}{%
    \immediate\write\@auxout
        {\string\providecommand*{\string\@gls@entry@count}[2]{}}%
    \forallglsentries{\@glsentry}{%
        \ifglsetused{\@glsentry}%
        {\immediate\write\@auxout
            {\string\@gls@entry@count{\@glsentry}{\glsetentrycurrcount{\@glsentry}}}}%
        {}%
    }%
}
\newcommand*{\@gls@entry@count}[2]{%
}
\newrobustcmd*{\cglset}{\@gls@hyp@opt\@cglset}
\newcommand*{\@cglset}[2][{}]{%

```

```

    \new@ifnextchar[{\@cgl@s@{#1}{#2}}{\@cgl@s@{#1}{#2} []}%
}
\def\@cgl@s@#1#2[#3]{%
  \GlossariesWarning{\string\cgl@s\space is defaulting to
    \string\gl@s\space since you haven't enabled entry counting}%
  \@gl@s@{#1}{#2}[#3]%
}
\newcommand*{\cgl@sformat}[2]{%
  \ifglshaslong{#1}{\gl@sentrylong{#1}}{\gl@sentryfirst{#1}}#2%
}
\newrobustcmd*{\cGls}{\@gl@s@hyp@opt\@cGls}
\glsmfuaddmap{\cgl@s}{\cGls}
\newcommand*{\@cGls}[2] []{%
  \new@ifnextchar[{\@cGls@{#1}{#2}}{\@cGls@{#1}{#2} []}%
}
\def\@cGls@#1#2[#3]{%
  \GlossariesWarning{\string\cGls\space is defaulting to
    \string\Gls\space since you haven't enabled entry counting}%
  \@Gls@{#1}{#2}[#3]%
}
\newcommand*{\cGlsformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cgl@spl}{\@gl@s@hyp@opt\@cgl@spl}
\newcommand*{\@cgl@spl}[2] []{%
  \new@ifnextchar[{\@cgl@spl@{#1}{#2}}{\@cgl@spl@{#1}{#2} []}%
}
\def\@cgl@spl@#1#2[#3]{%
  \GlossariesWarning{\string\cgl@spl\space is defaulting to
    \string\gl@spl\space since you haven't enabled entry counting}%
  \@gl@spl@{#1}{#2}[#3]%
}
\newcommand*{\cgl@splformat}[2]{%
  \ifglshaslong{#1}{\gl@sentrylongpl{#1}}{\gl@sentryfirstplural{#1}}#2%
}
\newrobustcmd*{\cGlspl}{\@gl@s@hyp@opt\@cGlspl}
\glsmfuaddmap{\cgl@spl}{\cGlspl}
\newcommand*{\@cGlspl}[2] []{%
  \new@ifnextchar[{\@cGlspl@{#1}{#2}}{\@cGlspl@{#1}{#2} []}%
}
\def\@cGlspl@#1#2[#3]{%
  \GlossariesWarning{\string\cGlspl\space is defaulting to
    \string\Glspl\space since you haven't enabled entry counting}%
  \@Glspl@{#1}{#2}[#3]%
}
\newcommand*{\cGlsplformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
}
\newcommand*{\loadglsentries}[2] [\@gl@s@default]{%
  \let\@gl@s@default\gl@sdefaulttype

```

```

\def\glsdefaulttype{#1}\input{#2}%
\let\glsdefaulttype\@gls@default
}
\@onlypreamble{\loadglsentries}
\newcommand*\glsformat[1]{#1}
\newcommand*\glsentryfmt{\glsentryfmt}
\newcommand*\glsentryfirst{\glsentryfirst}
\ifdefempty\glsentryfirst
{%
\glsifplural
{%
\glsupercase
{%
\ifglsused\glslabel
{%
\glsentryplural{\glslabel}\glsinsert
}%
{%
\glsentryfirstplural{\glslabel}\glsinsert
}%
}%
}%
\ifglsused\glslabel
{%
\Glsentryplural{\glslabel}\glsinsert
}%
{%
\Glsentryfirstplural{\glslabel}\glsinsert
}%
}%
}%
\ifglsused\glslabel
{%
\glsupercase{\glsentryplural{\glslabel}\glsinsert}%
}%
{%
\glsupercase{\glsentryfirstplural{\glslabel}\glsinsert}%
}%
}%
}%
\glsupercase
{%
\ifglsused\glslabel
{%
\glsentrytext{\glslabel}\glsinsert
}%
{%
\glsentryfirst{\glslabel}\glsinsert
}%
}

```

```

}%
{%
  \ifglused\glslabel
  {%
    \Glsentrytext{\glslabel}\glsinsert
  }%
  {%
    \Glsentryfirst{\glslabel}\glsinsert
  }%
}%
{%
  \ifglused\glslabel
  {%
    \glsuppercase{\glsentrytext{\glslabel}\glsinsert}%
  }%
  {%
    \glsuppercase{\glsentryfirst{\glslabel}\glsinsert}%
  }%
}%
}%
}%
}%
\glscustomtext\glsinsert
}%
}
\newcommand*{\glsnacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \acronymfont{\glsentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \glsuppercase
            {\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert}%
        }%
      }%
    }%
  }%
  {%
    \glscapscase
    {%
      \acronymfont{\glsentryshort{\glslabel}}\glsinsert
    }%
    {%

```

```

        \acronymfont{\Glsentryshort{\glslabel}}\glsinsert
    }%
    {%
        \glsuppercase
        {\acronymfont{\glsentryshort{\glslabel}}\glsinsert}%
    }%
}%
}%
{%
    \glsifplural
    {%
        \glscapscase
        {%
            \genplacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \Genplacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \glsuppercase
            {\genplacrfullformat{\glslabel}{\glsinsert}}%
        }%
    }%
    {%
        \glscapscase
        {%
            \genacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \Genacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \glsuppercase
            {\genacrfullformat{\glslabel}{\glsinsert}}%
        }%
    }%
}%
}%
}%
\glscustomtext
}%
}
\newcommand*{\genacrfullformat}[2]{%
    \glsentrylong{#1}#2\space
    (\protect\firstacronymfont{\glsentryshort{#1}})%
}
\newcommand*{\Genacrfullformat}[2]{%
    \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
    \expandafter\glsentencecase\expandafter{\gls@text}%
}

```

```

\newcommand*\genplacrfullformat}[2]{%
  \glsentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glsentryshortpl{#1}})%
}
\newcommand*\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}
\define@key{glslink}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
  {%
    \def\@gls@counter{#1}%
  }%
}
\define@key{glslink}{format}{%
  \def\@glsnumberformat{#1}}
\newcommand*\@gls@glslink@hyper@update@hook{}
\define@boolkey{glslink}{hyper}[true]{\@gls@glslink@hyper@update@hook}
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\newcommand*\@gls@save@glslocal{%
  \let\if@org@KV@glslink@local\ifKV@glslink@local
}
\newcommand*\@gls@restore@glslocal{%
  \ifKV@glslink@local
  \let\@gls@do@glsunset\glslocalunset
  \else
  \let\@gls@do@glsunset\glsunset
  \fi
  \let\ifKV@glslink@local\if@org@KV@glslink@local
}
\newcommand*\@gls@do@glsunset}[1]{\glsunset{#1}}
\define@boolkey{glslink}{local}[true]{}
\@gls@save@glslocal
\newcommand*\glslinkvar}[3]{#1}
\newcommand*\@gls@hyp@opt}[1]{%
  \let\glslinkvar\@firstofthree
  \let\@gls@hyp@opt@cs#1\relax
  \@ifstar{\s@gls@hyp@opt}%
  {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*\s@gls@hyp@opt}[1][]{%
  \let\glslinkvar\@secondofthree

```



```

\@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1][{}]{%
\let\glslinkvar\@thirdofthree
\@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
\@gls@hyp@opt\@gls@link
}
\newcommand*{\@gls@link}[3][{}]{%
\glsdoifexistsordo{#2}%
{%
\let\do@gls@link@checkfirsthyper\relax
\@gls@link[#1]{#2}{#3}%
}%
\glstextformat{#3}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Glslink}{%
\@gls@hyp@opt\@Gls@link
}
\newcommand*{\@Gls@link}[3][{}]{%
\@gls@link[#1]{#2}{\glssentencecase{#3}}%
}
\glsmfuaddmap{\glslink}{\Glslink}
\newcommand*{\glspostlinkhook}{}
\newcommand*{\@gls@link@checkfirsthyper}{%
\ifglsused{\glslabel}%
{%
}%
{%
\gls@checkisacronymlist\glstype
\ifglshyperfirst
\if@glsisacronymlist
\ifglsacrfootnote
\KV@glslink@hyperfalse
\fi
\fi
\else
\KV@glslink@hyperfalse
\fi
}%
\glslinkcheckfirsthyperhook
}
\newcommand*{\glslinkcheckfirsthyperhook}{}
\newcommand*{\glslinkpostsetkeys}{}
\newcommand{\glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}
\newcommand*{\do@glsdisablehyperinlist}{%
\expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
{\KV@glslink@hyperfalse}{}%
}
}

```

```

\newcommand*{\@gls@setdefault@glslink@opts}{%
\def\@gls@link[#1]#2#3{%
  \leavevmode
  \protected@edef\glslabel{\glsdetoklabel{#2}}%
  \def\@gls@link@opts{#1}%
  \let\@gls@link@label\glslabel
  \def\@glsnumberformat{glsnumberformat}%
  \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  \protected@edef\glstype{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
  \@gls@save@glslocal
  \@gls@setdefault@glslink@opts
  \do@glsdisablehyperinlist
  \do@gls@link@checkfirsthyper
  \setkeys{glslink}{#1}%
  \glslinkpostsetkeys
  \@gls@saveentrycounter
  \@gls@setsort{\glslabel}%
  \glsencapwrcontent{\@do@wrglossary{#2}}%
  \ifKV@glslink@hyper
    \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
  \else
    \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
  \fi
  \let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
  \@gls@restore@glslocal
}
\newcommand*{\glsencapwrcontent}[1]{#1}
\newcommand*{\glolinkprefix}{glo:}
\def\glsentrycounter{\glscounter}%
\newcommand*{\@gls@saveentrycounter}{%
\def\@gls@Hcounter{%
\ifthenelse{\equal{\@gls@counter}{equation}}%
{
\ifcsundef{xatlevel@}%
{%
\edef\theglsentrycounter{\expandafter\noexpand
\csname the\@gls@counter\endcsname}%
}%
{%
\ifx\xatlevel@\@empty
\edef\theglsentrycounter{\expandafter\noexpand
\csname the\@gls@counter\endcsname}%
\else
\savecounters@
\advance\c@equation by 1\relax
\edef\theglsentrycounter{\csname the\@gls@counter\endcsname}%
\ifcsundef{theH\@gls@counter}%
{%
\def\@gls@Hcounter{\theglsentrycounter}%
}
}
}
}

```

```

    }%
    {%
        \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
    }%
    \protected@edef\theHglentrycounter{\@gls@Hcounter}%
    \restorecounters@
\fi
}%
}%
{%
    \edef\theglentrycounter{\expandafter\noexpand
        \csname the\@gls@counter\endcsname}%
}%
\ifx\@gls@Hcounter\@empty
    \ifcsundef{theH\@gls@counter}%
    {%
        \def\theHglentrycounter{\theglentrycounter}%
    }%
    {%
        \protected@edef\theHglentrycounter{\expandafter\noexpand
            \csname theH\@gls@counter\endcsname}%
    }%
\fi
}
\def\@set@glo@numformat#1#2#3#4{%
    \expandafter\@glo@check@mkidxrangear#3\@nil
    \protected@edef#1{%
        \@glo@prefix setentrycounter[#4]{#2}%
        \expandafter\string\csname\@glo@suffix\endcsname
    }%
    \@gls@checkmkidxchars#1%
}
\def\@glo@check@mkidxrangear#1#2\@nil{%
\if#1(\relax
    \def\@glo@prefix{(%}
    \if\relax#2\relax
        \def\@glo@suffix{glsnumberformat}%
    \else
        \def\@glo@suffix{#2}%
    \fi
\else
    \if#1)\relax
        \def\@glo@prefix{)%}
    \if\relax#2\relax
        \def\@glo@suffix{glsnumberformat}%
    \else
        \def\@glo@suffix{#2}%
    \fi
\else
    \def\@glo@prefix{)\def\@glo@suffix{#1#2}%

```

```

\fi
\fi}
\newcommand*{\@gls@escbsdq}[1]{%
  \def\@gls@checkedmkidx{}%
  \let\gls@xdystring=#1\relax
  \@onelevel@sanitize\gls@xdystring
  \edef\do@gls@xdycheckbackslash{%
    \noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
    \@backslashchar\@backslashchar\noexpand\null}%
  \do@gls@xdycheckbackslash
  \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
  \def\@gls@checkedmkidx{}%
  \expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\null
  \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
  \ifglsesclocations
    \for\@gls@tmp:=\gls@protected@pagefmts\do
    {%
      \edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\ \expandonce\@gls@tmp}%
      \@onelevel@sanitize\@gls@sanitized@tmp
      \edef\gls@dostsubst{%
        \noexpand\DTLsubstituteall\noexpand\gls@xdystring
        {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
      }%
      \gls@dostsubst
    }%
  \fi
  \let#1=\gls@xdystring
}
\newcommand{\@gls@checkmkidxchars}[1]{%
  \ifglsxindy
    \@gls@escbsdq{#1}%
  \else
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@checkquote#1\@nil""\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@checkescquote#1\@nil""\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@checkescactual#1\@nil\?\?\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@checkactual#1\@nil??\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@checkbar#1\@nil||\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@checkescbar#1\@nil\\|\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
  \fi
}

```

```

\def\@gls@checkedmkidx{%
\expandafter\@gls@checklevel#1\@nil!!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkesclevel#1\@nil\!\!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\edef\@do@gls@checkesc{%
\noexpand\@gls@checkesc\expandonce{#1}\noexpand\@nil
\backslashchar\@backslashchar\noexpand\null
}%
\do@gls@checkesc
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\fi
}
\def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
\newtoks\@gls@tmpb
\def\@gls@checkquote#1"#2"#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
\def\@gls@checkquote{\@gls@checkquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@quotechar}%
\ifx\null#3\null
\def\@gls@checkquote{\@gls@checkquote#2""\null}%
\else
\def\@gls@checkquote{\@gls@checkquote#2"#3\null}%
\fi
\fi
\@gls@checkquote
}
\def\@gls@checkescquote#1"#2"#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\@\@gls@quotechar

```

```

        \@gls@quotechar\string"\@gls@quotechar}%
\def\@gls@checkescquote{\@gls@checkescquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar}%
\ifx\null#3\null
\def\@gls@checkescquote{\@gls@checkescquote#2"\@gls@quotechar}%
\else
\def\@gls@checkescquote{\@gls@checkescquote#2"#3\null}%
\fi
\fi
\@gls@checkescquote
}
\def\@gls@checkescactual#1\?#2\?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar
\@gls@quotechar\string"\@gls@actualchar}%
\def\@gls@checkescactual{\@gls@checkescactual#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar}%
\ifx\null#3\null
\def\@gls@checkescactual{\@gls@checkescactual#2\?\?\null}%
\else
\def\@gls@checkescactual{\@gls@checkescactual#2\?#3\null}%
\fi
\fi
\@gls@checkescactual
}
\def\@gls@checkescbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@encapchar
\@gls@quotechar\string"\@gls@encapchar}%
\def\@gls@checkescbar{\@gls@checkescbar#3\null}%

```

```

\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\\"\@gls@encapchar}%
\ifx\@null#3\@null
\def\@@gls@checkeschar{\@gls@checkeschar#2\|\|\@null}%
\else
\def\@@gls@checkeschar{\@gls@checkeschar#2\|#3\@null}%
\fi
\fi
\@gls@checkeschar
}
\def\@gls@checkescllevel#1\!#2\!#3\@null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\@null#2\@null
\ifx\@null#3\@null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkescllevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\\"\@gls@levelchar
\@gls@quotechar\string\\"\@gls@levelchar}%
\def\@@gls@checkescllevel{\@gls@checkescllevel#3\@null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\\"\@gls@levelchar}%
\ifx\@null#3\@null
\def\@@gls@checkescllevel{\@gls@checkescllevel#2\!\!\@null}%
\else
\def\@@gls@checkescllevel{\@gls@checkescllevel#2\!#3\@null}%
\fi
\fi
\@gls@checkescllevel
}
\def\@gls@checkbar#1|#2|#3\@null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\@null#2\@null
\ifx\@null#3\@null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@@gls@checkbar{\@gls@checkbar#3\@null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@

```

```

    \@gls@quotechar\@gls@encapchar}%
  \ifx\null#3\null
    \def\@gls@checkbar{\@gls@checkbar#2||\null}%
  \else
    \def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
  \fi
\fi
\@gls@checkbar
}
\def\@gls@checklevel#1!#2!#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checklevel{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
      \def\@gls@checklevel{\@gls@checklevel#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@gls@levelchar}%
    \ifx\null#3\null
      \def\@gls@checklevel{\@gls@checklevel#2!!\null}%
    \else
      \def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
    \fi
  \fi
  \@gls@checklevel
}
\def\@gls@checkactual#1?#2?#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkactual{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
      \def\@gls@checkactual{\@gls@checkactual#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@gls@actualchar}%
    \ifx\null#3\null
      \def\@gls@checkactual{\@gls@checkactual#2??\null}%
    \else

```



```

        \def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
        \fi
        \fi
        \@gls@checkactual
    }
    \bgroup
    \catcode'\|0\relax
    \catcode'\|12\relax
    \gdef\@gls@checkesc#1\#2\#3\null{%
        \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
        \toks@={#1}%
        \ifx\null#2\null
            \ifx\null#3\null
                \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
                \def\@gls@checkesc{\relax}%
            \else
                \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
                    \@gls@quotechar\@backslashchar\@gls@quotechar\@backslashchar}%
                \def\@gls@checkesc{\@gls@checkesc#3\null}%
            \fi
        \else
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
                \@gls@quotechar\@backslashchar}%
            \ifx\null#3\null
                \def\@gls@checkesc{\@gls@checkesc#2\| \null}%
            \else
                \def\@gls@checkesc{\@gls@checkesc#2\#3\null}%
            \fi
        \fi
        \@gls@checkesc
    }
    \egroup
    \def\@gls@xdycheckquote#1"#2"#3\null{%
        \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
        \toks@={#1}%
        \ifx\null#2\null
            \ifx\null#3\null
                \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
                \def\@gls@xdycheckquote{\relax}%
            \else
                \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
                    \string\string\}%
                \def\@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
            \fi
        \else
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
                \string\}%
            \ifx\null#3\null
                \def\@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
            \else

```

```

        \def\@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
        \fi
        \fi
    \@gls@xdycheckquote
}
\edef\def@gls@xdycheckbackslash{%
\noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
##2\@backslashchar##3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter
{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={##1}%
\noexpand\ifx\noexpand\null##2\noexpand\null
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar}%
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
\@backslashchar\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
##3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@xdycheckbackslash
}%
}
\def@gls@xdycheckbackslash

\newlength\gls@tmplen
\newcommand*\glsdohypertarget}[2]{%
\glsdoshowtarget{#1}%
{
\glsmeasureheight{\gls@tmplen}{#2}%
\glsdohypertargethook{#1}{#2}%
\raisebox{\gls@tmplen}{\hypertarget{#1}{}}%
#2%
}
}

```

```

    }%
  }
  \newcommand*\glsdohypertargethook}[2]{}
  \NewDocumentCommand\glslabelhypertarget{mm}{%
    \protected@write\@auxout
      {\glslabelhypertargetdefs}%
      {%
        \string\newlabel
          {\glslabelhypertargetprefix#1}%
          {%
            {\glslabelhypertargetvalue}{\thepage}{#2}{#1}{}%
          }%
        }%
      }%
  }
  \ExplSyntaxOn
  \newcommand\glslabelhypertargetvalue{
    \bool_lazy_and:nnT
      { \tl_if_exist_p:N \glscurrententrylabel }
      { \bool_not_p:n { \tl_if_empty_p:N \glscurrententrylabel } }
    {
      \glsentryname \glscurrententrylabel
    }
  }
  \ExplSyntaxOff
  \newcommand*\glslabelhypertargetprefix{}
  \newcommand*\glslabelhypertargetdefs{%
    \let\glossentryname\glsentryname
    \let\glossentrydesc\glsentrydesc
    \let\glossentrysymbol\glsentrysymbol
    \let\Glossentryname\glsentryname
    \let\Glossentrydesc\glsentrydesc
    \let\Glossentrysymbol\glsentrysymbol
    \let\glossentrynameother\@gls@entry@field
  }
  \newcommand*\glsdohyperlink}[2]{%
    \glsdoshowtarget{#1}%
    {%
      \glsdohyperlinkhook{#1}{#2}%
      \hyperlink{#1}{#2}%
    }%
  }
  \newcommand*\glsdohyperlinkhook}[2]{}
  \newcommand*\glsdonohyperlink}[2]{#2}
  \ifcsundef{hyperlink}%
  {%
    \let\@glslink\glsdonohyperlink
  }%
  {%
    \let\@glslink\glsdohyperlink
  }
}

```

```

\ifcsundef{hypertarget}%
{%
  \let\@glstarget\@secondoftwo
}%
{%
  \let\@glstarget\glstdohypertarget
}
\newcommand{\glsdisablehyper}{%
  \KV@glslink@hyperfalse
  \let\@glslink\glsdonohyperlink
  \let\@glstarget\@secondoftwo
}
\newcommand{\glsenablehyper}{%
  \KV@glslink@hypertrue
  \let\@glslink\glstdohyperlink
  \let\@glstarget\glstdohypertarget
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}
\newcommand*{\@gls}[2][{}]{%
  \new@ifnextchar[{\@gls@#1}{#2}]{\@gls@#1}{#2}[{}]}%
}
\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glsescapscase\@firstofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\gls@type @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \@gls@do@glsunset{#2}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
\glsmfuaddmap{\gls}{\Gls}
\newcommand*{\@Gls}[2][{}]{%
  \new@ifnextchar[{\@Gls@#1}{#2}]{\@Gls@#1}{#2}[{}]}%
}
\def\@Gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glsescapscase\@secondofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
  }%
}

```

```

\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}
\glsmfublocker{\GLS}
\newcommand*{\@GLS}[2] []{%
\new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2} []}%
}
\def\@GLS@#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@secondoftwo
\let\gls@scaps@case\@thirdofthree
\let\gls@customtext\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\glspl}{\@gls@hyp@opt\@glspl}
\newcommand*{\@glspl}[2] []{%
\new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2} []}%
}
\def\@glspl@#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@firstoftwo
\let\gls@scaps@case\@firstofthree
\let\gls@customtext\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
\glsmfuaddmap{\glspl}{\Glspl}
\newcommand*{\@Glspl}[2] []{%
\new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2} []}%
}
\def\@Glspl@#1#2[#3]{%
\glsdoifexists{#2}%

```

```

{%
  \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
  \let\gl@ifplural\@firstoftwo
  \let\gl@scapscase\@secondofthree
  \let\gl@customtext\@empty
  \def\gl@insert{#3}%
  \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
  \@gl@link[#1]{#2}{\@glo@text}%
  \@gl@do@gl@unset{#2}%
}%
\gl@postlinkhook
}
\newrobustcmd*{\GLSp1}{\@gl@hyp@opt\@GLSp1}
\gl@mf@blocker{\GLSp1}
\newcommand*{\@GLSp1}[2][{}]{%
  \new@ifnextchar[{\@GLSp1@{#1}{#2}}{\@GLSp1@{#1}{#2}[]}%
}
\def\@GLSp1@#1#2[#3]{%
  \gl@do@ifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\gl@ifplural\@firstoftwo
    \let\gl@scapscase\@thirdofthree
    \let\gl@customtext\@empty
    \def\gl@insert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \@gl@do@gl@unset{#2}%
  }%
  \gl@postlinkhook
}
\newrobustcmd*{\gl@disp}{\@gl@hyp@opt\@gl@disp}
\newcommand*{\@gl@disp}[3][{}]{%
  \gl@do@ifexists{#2}{%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\gl@ifplural\@secondoftwo
    \let\gl@scapscase\@firstofthree
    \def\gl@customtext{#3}%
    \def\gl@insert{}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \@gl@do@gl@unset{#2}%
  }%
  \gl@postlinkhook
}
\newrobustcmd*{\Glsdisp}{\@gl@hyp@opt\@Glsdisp}
\newcommand*{\@Glsdisp}[3][{}]{%
  \@gl@disp[#1]{#2}{\gl@sentencecase{#3}}%
}
\gl@mf@uaddmap{\gl@disp}{\Glsdisp}

```

```

\newcommand*{\@gls@link@nocheckfirsthyper}{}
\newcommand{\@gls@field@link}[3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \@gls@link{#1}{#2}{#3}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\gls@hyp@opt\@gls@link}
\newcommand*{\@gls@link}[2] [] {%
  \new@ifnextchar[{\@gls@link@{#1}{#2}}{\@gls@link@{#1}{#2} []}]
\def\@gls@link@{#1}{#2}{#3}{%
  \@gls@field@link{#1}{#2}{\glsentrytext{#2}{#3}}%
}
\newrobustcmd*{\GLS@hyp@opt\@GLS@link}
\glsmfublocker{\@GLS@link}
\newcommand*{\@GLS@link}[2] [] {%
  \new@ifnextchar[{\@GLS@link@{#1}{#2}}{\@GLS@link@{#1}{#2} []}]
\def\@GLS@link@{#1}{#2}{#3}{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrytext{#2}{#3}}}%
}
\newrobustcmd*{\Gls@hyp@opt\@Gls@link}
\glsmfuaddmap{\@Gls@link}
\newcommand*{\@Gls@link}[2] [] {%
  \new@ifnextchar[{\@Gls@link@{#1}{#2}}{\@Gls@link@{#1}{#2} []}]
\def\@Gls@link@{#1}{#2}{#3}{%
  \@gls@field@link{#1}{#2}{\Glsentrytext{#2}{#3}}%
}
\newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
\newcommand*{\@glsfirst}[2] [] {%
  \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2} []}]
\def\@glsfirst@{#1}{#2}{#3}{%
  \@gls@field@link{#1}{#2}{\glsentryfirst{#2}{#3}}%
}
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
\glsmfuaddmap{\@Glsfirst}
\newcommand*{\@Glsfirst}[2] [] {%
  \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2} []}]
\def\@Glsfirst@{#1}{#2}{#3}{%
  \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}{#3}}%
}
\newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
\glsmfublocker{\@GLSfirst}
\newcommand*{\@GLSfirst}[2] [] {%
  \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2} []}]
\def\@GLSfirst@{#1}{#2}{#3}{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryfirst{#2}{#3}}}%
}
\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}

```

```

\newcommand*{\@glsplural}[2] []{%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} []}]
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\glsmfuaddmap{\glsplural}{\Glsplural}
\newcommand*{\@Glsplural}[2] []{%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} []}]
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
\newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
\glsmfublocker{\GLSplural}
\newcommand*{\@GLSplural}[2] []{%
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} []}]
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentryplural{#2}#3}}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{\@glsfirstplural}[2] []{%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} []}]
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
\glsmfuaddmap{\glsfirstplural}{\Glsfirstplural}
\newcommand*{\@Glsfirstplural}[2] []{%
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2} []}]
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
\glsmfublocker{\GLSfirstplural}
\newcommand*{\@GLSfirstplural}[2] []{%
  \new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} []}]
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentryfirstplural{#2}#3}}%
}
\newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
\newcommand*{\@glsname}[2] []{%
  \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2} []}]
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryname{#2}#3}%
}
\newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}
\glsmfuaddmap{\glsname}{\Glsname}
\newcommand*{\@Glsname}[2] []{%
  \new@ifnextchar[{\@Glsname@{#1}{#2}}{\@Glsname@{#1}{#2} []}]
\def\@Glsname@#1#2[#3]{%

```



```

    \@gls@field@link{#1}{#2}{\Glsentryname{#2}#3}%
}
\newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}
\glsmfublocker{\GLSname}
\newcommand*{\@GLSname}[2] [] {%
    \new@ifnextchar [{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}]
}
\def\@GLSname@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryname{#2}#3}}%
}
\newrobustcmd*{\glsdesc}{\@gls@hyp@opt\@glsdesc}
\newcommand*{\@glsdesc}[2] [] {%
    \new@ifnextchar [{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2} []}]
}
\def\@glsdesc@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsentrydesc{#2}#3}%
}
\newrobustcmd*{\Glsdesc}{\@gls@hyp@opt\@Glsdesc}
\glsmfuaddmap{\glsdesc}{\Glsdesc}
\newcommand*{\@Glsdesc}[2] [] {%
    \new@ifnextchar [{\@Glsdesc@{#1}{#2}}{\@Glsdesc@{#1}{#2} []}]
}
\def\@Glsdesc@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%
}
\newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\glsmfublocker{\GLSdesc}
\newcommand*{\@GLSdesc}[2] [] {%
    \new@ifnextchar [{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}]
}
\def\@GLSdesc@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrydesc{#2}#3}}%
}
\newrobustcmd*{\glsdescplural}{\@gls@hyp@opt\@glsdescplural}
\newcommand*{\@glsdescplural}[2] [] {%
    \new@ifnextchar [{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}]
}
\def\@glsdescplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\Glsdescplural}{\@gls@hyp@opt\@Glsdescplural}
\glsmfuaddmap{\glsdescplural}{\Glsdescplural}
\newcommand*{\@Glsdescplural}[2] [] {%
    \new@ifnextchar [{\@Glsdescplural@{#1}{#2}}{\@Glsdescplural@{#1}{#2} []}]
}
\def\@Glsdescplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\glsmfublocker{\GLSdescplural}
\newcommand*{\@GLSdescplural}[2] [] {%
    \new@ifnextchar [{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}]
}
\def\@GLSdescplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrydescplural{#2}#3}}%
}
\newrobustcmd*{\glsymbol}{\@gls@hyp@opt\@glsymbol}

```

```

\newcommand*{\@glssymbol}[2] [] {%
  \new@ifnextchar [{\@glssymbol@{#1}{#2}}{\@glssymbol@{#1}{#2} []}]
\def\@glssymbol@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glstentrysymbol{#2}#3}%
}
\newrobustcmd*{\Glsymbol}{\@gls@hyp@opt\@Glsymbol}
\glsmfuaddmap{\glssymbol}{\Glsymbol}
\newcommand*{\@GLssymbol}[2] [] {%
  \new@ifnextchar [{\@GLssymbol@{#1}{#2}}{\@GLssymbol@{#1}{#2} []}]
\def\@GLssymbol@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glstentrysymbol{#2}#3}%
}
\newrobustcmd*{\GLSsymbol}{\@gls@hyp@opt\@GLSsymbol}
\glsmfublocker{\GLSsymbol}
\newcommand*{\@GLSsymbol}[2] [] {%
  \new@ifnextchar [{\@GLSsymbol@{#1}{#2}}{\@GLSsymbol@{#1}{#2} []}]
\def\@GLSsymbol@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glssuppercase{\glstentrysymbol{#2}#3}}%
}
\newrobustcmd*{\glssymbolplural}{\@gls@hyp@opt\@glssymbolplural}
\newcommand*{\@glssymbolplural}[2] [] {%
  \new@ifnextchar [{\@glssymbolplural@{#1}{#2}}{\@glssymbolplural@{#1}{#2} []}]
\def\@glssymbolplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glstentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
\glsmfuaddmap{\glssymbolplural}{\GLSsymbolplural}
\newcommand*{\@GLSsymbolplural}[2] [] {%
  \new@ifnextchar [{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2} []}]
\def\@GLSsymbolplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glstentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
\glsmfublocker{\GLSsymbolplural}
\newcommand*{\@GLSsymbolplural}[2] [] {%
  \new@ifnextchar [{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2} []}]
\def\@GLSsymbolplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glssuppercase{\glstentrysymbolplural{#2}#3}}%
}
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
\newcommand*{\@glsuseri}[2] [] {%
  \new@ifnextchar [{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2} []}]
\def\@glsuseri@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glstentryuseri{#2}#3}%
}
\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
\glsmfuaddmap{\glsuseri}{\Glsuseri}
\newcommand*{\@Glsuseri}[2] [] {%
  \new@ifnextchar [{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2} []}]
\def\@Glsuseri@#1#2[#3] {%

```

```

    \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\glsmfublocker{\GLSuseri}
\newcommand*{\@GLSuseri}[2][\%
    \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2}[]}]
\def\@GLSuseri@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseri{#2}#3}}%
}
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
\newcommand*{\@glsuserii}[2][\%
    \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2}[]}]
\def\@glsuserii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryuserii{#2}#3}%
}
\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
\glsmfuaddmap{\glsuserii}{\Glsuserii}
\newcommand*{\@Glsuserii}[2][\%
    \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2}[]}]
\def\@Glsuserii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\GLSuserii}{\@gls@hyp@opt\@GLSuserii}
\glsmfublocker{\GLSuserii}
\newcommand*{\@GLSuserii}[2][\%
    \new@ifnextchar[{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2}[]}]
\def\@GLSuserii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuserii{#2}#3}}%
}
\newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}
\newcommand*{\@glsuseriii}[2][\%
    \new@ifnextchar[{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2}[]}]
\def\@glsuseriii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
\glsmfuaddmap{\glsuseriii}{\Glsuseriii}
\newcommand*{\@Glsuseriii}[2][\%
    \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2}[]}]
\def\@Glsuseriii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\GLSuseriii}{\@gls@hyp@opt\@GLSuseriii}
\glsmfublocker{\GLSuseriii}
\newcommand*{\@GLSuseriii}[2][\%
    \new@ifnextchar[{\@GLSuseriii@{#1}{#2}}{\@GLSuseriii@{#1}{#2}[]}]
\def\@GLSuseriii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseriii{#2}#3}}%
}
\newrobustcmd*{\glsuseriv}{\@gls@hyp@opt\@glsuseriv}

```

```

\newcommand*{\@glsuseriv}[2] []{%
  \new@ifnextchar[{\@glsuseriv@{#1}{#2}}{\@glsuseriv@{#1}{#2} []}]
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
\glsmfuaddmap{\glsuseriv}{\Glsuseriv}
\newcommand*{\@Glsuseriv}[2] []{%
  \new@ifnextchar[{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2} []}]
\def\@Glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\GLSuseriv}{\@gls@hyp@opt\@GLSuseriv}
\glsmfublocker{\GLSuseriv}
\newcommand*{\@GLSuseriv}[2] []{%
  \new@ifnextchar[{\@GLSuseriv@{#1}{#2}}{\@GLSuseriv@{#1}{#2} []}]
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseriv{#2}#3}}%
}
\newrobustcmd*{\glsuserv}{\@gls@hyp@opt\@glsuserv}
\newcommand*{\@glsuserv}[2] []{%
  \new@ifnextchar[{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2} []}]
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
\glsmfublocker{\GLSuserv}
\newcommand*{\@GLSuserv}[2] []{%
  \new@ifnextchar[{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2} []}]
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuserv{#2}#3}}%
}
\newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
\newcommand*{\@glsuservi}[2] []{%
  \new@ifnextchar[{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2} []}]
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
\newrobustcmd*{\Glsuservi}{\@gls@hyp@opt\@Glsuservi}
\glsmfuaddmap{\glsuservi}{\Glsuservi}
\newcommand*{\@Glsuservi}[2] []{%
  \new@ifnextchar[{\@Glsuservi@{#1}{#2}}{\@Glsuservi@{#1}{#2} []}]
\def\@Glsuservi@#1#2[#3]{%

```

```

    \@gls@field@link{#1}{#2}{\Glsentryuservi{#2}#3}%
}
\newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
\glsmfublocker{\GLSuservi}
\newcommand*{\@GLSuservi}[2] [] {%
  \new@ifnextchar [{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2} []}]
\def\@GLSuservi@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuservi{#2}#3}}%
}
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\ns@acrshort}
\newcommand*{\ns@acrshort}[2] [] {%
  \new@ifnextchar [{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2} []}]
}
\def\@acrshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrshort}{\@gls@hyp@opt\ns@Acrshort}
\glsmfuaddmap{\Acrshort}{\Acrshort}
\newcommand*{\ns@Acrshort}[2] [] {%
  \new@ifnextchar [{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2} []}]
}
\def\@Acrshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshort}{\@gls@hyp@opt\ns@ACRshort}
\glsmfublocker{\ACRshort}
\newcommand*{\ns@ACRshort}[2] [] {%

```

```

\new@ifnextchar [{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}%
}
\def\@ACRshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\glsuppercase{\acronymfont{\glsentryshort{#2}}#3}%
}%
\@gls@link[#1]{#2}{\csname gls@\gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\acrshortpl}{\@gls@hyp@opt\@ns@acrshortpl}
\newcommand*{\ns@acrshortpl}[2] []{%
\new@ifnextchar [{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} []}%
}
\def\@acrshortpl#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glsapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glsentryshortpl{#2}}#3%
}%
\@gls@link[#1]{#2}{\csname gls@\gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\@ns@Acrshortpl}
\glsmfuaddmap{\Acrshortpl}{\Acrshortpl}
\newcommand*{\ns@Acrshortpl}[2] []{%
\new@ifnextchar [{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2} []}%
}
\def\@Acrshortpl#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glsapscase\@secondofthree
\let\glsinsert\@empty
\def\glscustomtext{%

```

```

        \acronymfont{\Glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}
\glspostlinkhook
}
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\ns@ACRshortpl}
\glsmfublocker{\ACRshortpl}
\newcommand*{\ns@ACRshortpl}[2] []{%
    \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2} []}%
}
\def\@ACRshortpl#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \def\glslabel{#2}%
        \let\glsifplural\@firstoftwo
        \let\glscapscase\@thirdofthree
        \let\glsinsert\@empty
        \def\glscustomtext{%
            \glsuppercase{\acronymfont{\Glsentryshortpl{#2}}#3}%
        }%
        \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
    }%
    \glspostlinkhook
}
\newrobustcmd*{\acrlong}{\@gls@hyp@opt\ns@acrlong}
\newcommand*{\ns@acrlong}[2] []{%
    \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2} []}%
}
\def\@acrlong#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \def\glslabel{#2}%
        \let\glsifplural\@secondoftwo
        \let\glscapscase\@firstofthree
        \let\glsinsert\@empty
        \def\glscustomtext{%
            \glsentrylong{#2}#3%
        }%
        \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
    }%
    \glspostlinkhook
}
\newrobustcmd*{\Acrlong}{\@gls@hyp@opt\ns@Acrlong}
\glsmfuaddmap{\Acrlong}{\Acrlong}
\newcommand*{\ns@Acrlong}[2] []{%
    \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2} []}%
}

```

```

\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@secondofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \glsentrylong{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \gls post link hook
}
\newrobustcmd*\@ACRlong{\@gls@hyp@opt\ns@ACRlong}
\glsmfublocker{\@ACRlong}
\newcommand*\@ns@ACRlong}[2][{}]{%
  \new@ifnextchar[{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2}[]}%
}
\def\@ACRlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@thirdofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \glsuppercase{\glsentrylong{#2}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \gls post link hook
}
\newrobustcmd*\@acrlongpl{\@gls@hyp@opt\ns@acrlongpl}
\newcommand*\@ns@acrlongpl}[2][{}]{%
  \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2}[]}%
}
\def\@acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\gls caps case\@firstofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \glsentrylongpl{#2}#3%
    }%
  }%
}

```



```

    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\ns@Acrlongpl}
\glsmfuaddmap{\Acrlongpl}{\Acrlongpl}
\newcommand*{\ns@Acrlongpl}[2][]{%
  \new@ifnextchar[{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2} []}%
}
\def\@Acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\ns@ACRlongpl}
\glsmfublocker{\ACRlongpl}
\newcommand*{\ns@ACRlongpl}[2][]{%
  \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2} []}%
}
\def\@ACRlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsuppercase{\glsentrylongpl{#2}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newcommand*{\@gls@entry@field}[2]{%
  \csuse{glo@glsdetoklabel{#1}@#2}%
}
\newcommand*{\glsletentryfield}[3]{%
  \letcs{#1}{glo@glsdetoklabel{#2}@#3}%
}

```

```

\newrobustcmd*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@#2}%
    \ifdef\@glo@text
    {%
      \expandafter\glsentencecase\expandafter{\@glo@text}%
    }%
    {%
      ??\PackageError{glossaries}{The field '#2' doesn't exist for glossary
        entry '\glsdetoklabel{#1}'}{Check you have correctly spelt the entry
        label and the field name}%
    }%
  }%
  {%
    ??%
  }%
}
\newcommand{\@Gls@entry@field}[2]{%
  \glstexorpdfstring{\@Gls@entry@field{#1}{#2}}%
  {\MFUsentencecase{\@Gls@entry@field{#1}{#2}}}%
}
\ifdef\texorpdfstring
{\newcommand{\glstexorpdfstring}{\texorpdfstring}}
{\newcommand{\glstexorpdfstring}[2]{#1}}
\newcommand*{\Gls@entryname}[1]{\@Gls@entry@field{#1}{name}}
\newcommand*{\Gls@entryname}[1]{%
  \glstexorpdfstring{\protect\@Gls@entryname{#1}}%
  {\MFUsentencecase{\@Gls@entry@field{#1}{name}}}%
}
\glsmfuexcl{\Gls@entryname}
\newcommand*{\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
\newrobustcmd*{\@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \protected@edef\@glo@text{\@Gls@entry@field{#1}{name}}%
    \expandafter\glsentencecase\expandafter{\@glo@text}%
  }%
  {%
    \@Gls@entry@field{#1}{name}%
  }%
}
\newcommand*{\@org@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@name}%
    \expandafter\@gls@getbody\@glo@text{\@nil
    \expandafter\ifx\@gls@body\glsentrylong\relax

```

```

\expandafter\Glsentrylong\@gls@rest
\else
\expandafter\ifx\@gls@body\glsentryshort\relax
\expandafter\Glsentryshort\@gls@rest
\else
\expandafter\ifx\@gls@body\acronymfont\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\expandafter\glssentencecase\expandafter{\@glo@text}%
\fi
\fi
\fi
\fi
\fi
}%
{%
\@Gls@entry@field{#1}{name}%
}%
}
\newcommand*\glsentrydesc[1]{\@gls@entry@field{#1}{desc}}
\newcommand*\Glsentrydesc[1]{\@Glsentryfield{#1}{desc}}
\glsmfuexcl{\Glsentrydesc}
\newcommand*\glsentrydescplural[1]{%
\@gls@entry@field{#1}{descplural}}
}
\newcommand*\Glsentrydescplural[1]{%
\@Glsentryfield{#1}{descplural}}
}
\glsmfuexcl{\Glsentrydescplural}
\newcommand*\glsentrytext[1]{\@gls@entry@field{#1}{text}}
\newcommand*\Glsentrytext[1]{%
\@Glsentryfield{#1}{text}}
}
\glsmfuexcl{\Glsentrytext}
\newcommand*\glsentryplural[1]{%
\@gls@entry@field{#1}{plural}}
}
\newcommand*\Glsentryplural[1]{%
\@Glsentryfield{#1}{plural}}
}
\glsmfuexcl{\Glsentryplural}
\newcommand*\glsentrysymbol[1]{%

```

```

    \@gls@entry@field{#1}{symbol}%
  }
  \newcommand*{\Glsentrysymbol}[1]{%
    \@Glsentryfield{#1}{symbol}%
  }
  \glsmfuexcl{\Glsentrysymbol}
  \newcommand*{\glsentrysymbolplural}[1]{%
    \@gls@entry@field{#1}{symbolplural}%
  }
  \newcommand*{\Glsentrysymbolplural}[1]{%
    \@Glsentryfield{#1}{symbolplural}%
  }
  \glsmfuexcl{\Glsentrysymbolplural}
  \newcommand*{\glsentryfirst}[1]{%
    \@gls@entry@field{#1}{first}%
  }
  \newcommand*{\Glsentryfirst}[1]{%
    \@Glsentryfield{#1}{first}%
  }
  \glsmfuexcl{\Glsentryfirst}
  \newcommand*{\glsentryfirstplural}[1]{%
    \@gls@entry@field{#1}{firstpl}%
  }
  \newcommand*{\Glsentryfirstplural}[1]{%
    \@Glsentryfield{#1}{firstpl}%
  }
  \glsmfuexcl{\Glsentryfirstplural}
  \newcommand{\glscapitalisewords}[1]{%
    \@capitalisewords{#1}%
  }
  \newcommand*{\glsentrytitlecase}[2]{%
    \glstexorpdfstring
      {\@glsentrytitlecase{#1}{#2}}%
      {\MFUsentencecase{\@gls@entry@field{#1}{#2}}}%
  }
  \newrobustcmd*{\@glsentrytitlecase}[2]{%
    \glsdoifexists{#1}%
    {%
      \glsfieldfetch{#1}{#2}{\@gls@value}%
      \expandafter\glscapitalisewords\expandafter{\@gls@value}%
    }%
  }
  \newcommand*{\glsentrytype}[1]{\@gls@entry@field{#1}{type}}
  \newcommand*{\glsentrysort}[1]{%
    \@gls@entry@field{#1}{sort}%
  }
  \newcommand*{\glsentryparent}[1]{%
    \@gls@entry@field{#1}{parent}%
  }
  \newcommand*{\glsentryuseri}[1]{%

```

```

    \@gls@entry@field{#1}{useri}%
}
\newcommand*{\Glsentryuseri}[1]{%
    \@Glsentryfield{#1}{useri}%
}
\glsmfuexcl{\Glsentryuseri}
\newcommand*{\glsentryuserii}[1]{%
    \@gls@entry@field{#1}{userii}%
}
\newcommand*{\Glsentryuserii}[1]{%
    \@Glsentryfield{#1}{userii}%
}
\glsmfuexcl{\Glsentryuserii}
\newcommand*{\glsentryuseriii}[1]{%
    \@gls@entry@field{#1}{useriii}%
}
\newcommand*{\Glsentryuseriii}[1]{%
    \@Glsentryfield{#1}{useriii}%
}
\glsmfuexcl{\Glsentryuseriii}
\newcommand*{\glsentryuseriv}[1]{%
    \@gls@entry@field{#1}{useriv}%
}
\newcommand*{\Glsentryuseriv}[1]{%
    \@Glsentryfield{#1}{useriv}%
}
\glsmfuexcl{\Glsentryuseriv}
\newcommand*{\glsentryuserv}[1]{%
    \@gls@entry@field{#1}{userv}%
}
\newcommand*{\Glsentryuserv}[1]{%
    \@Glsentryfield{#1}{userv}%
}
\glsmfuexcl{\Glsentryuserv}
\newcommand*{\glsentryuservi}[1]{%
    \@gls@entry@field{#1}{uservi}%
}
\newcommand*{\Glsentryuservi}[1]{%
    \@Glsentryfield{#1}{uservi}%
}
\glsmfuexcl{\Glsentryuservi}
\newcommand*{\glsentryshort}[1]{\@gls@entry@field{#1}{short}}
\newcommand*{\Glsentryshort}[1]{%
    \@Glsentryfield{#1}{short}%
}
\glsmfuexcl{\Glsentryshort}
\newcommand*{\glsentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}
\newcommand*{\Glsentryshortpl}[1]{%
    \@Glsentryfield{#1}{shortpl}%
}
}

```

```

\glsmfueycl{\Glsentryshortpl}
\newcommand*{\glsentrylong}[1]{\@gls@entry@field{#1}{long}}
\newcommand*{\Glsentrylong}[1]{%
  \@Glsentryfield{#1}{long}%
}
\glsmfueycl{\Glsentrylong}
\newcommand*{\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}
\newcommand*{\Glsentrylongpl}[1]{%
  \@Glsentryfield{#1}{longpl}%
}
\glsmfueycl{\Glsentrylongpl}
\newcommand*{\glsentryfull}[1]{%
  \glsentrylong{#1} (\glsentryshort{#1})%
}
\newcommand*{\Glsentryfull}[1]{%
  \Glsentrylong{#1} (\glsentryshort{#1})%
}
\glsmfueycl{\Glsentryfull}
\newcommand*{\glsentryfullpl}[1]{%
  \acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newcommand*{\Glsentryfullpl}[1]{%
  \Glsentrylongpl{#1} (\glsentryshortpl{#1})%
}
\glsmfueycl{\Glsentryfullpl}
\newcommand*{\glsentrynumberlist}[1]{%
  \glsdoifexists{#1}%
  {%
    \@gls@entry@field{#1}{numberlist}%
  }%
}
\@ifpackageloaded{hyperref} {%
  \newcommand*{\glsdisplaynumberlist}[1]{%
    \GlossariesWarning
    {%
      \string\glsdisplaynumberlist\space
      doesn't work with hyperref.^^JUsing
      \string\glsentrynumberlist\space instead%
    }%
    \glsentrynumberlist{#1}%
  }%
}%
\newcommand*{\glsdisplaynumberlist}[1]{%
  \glsdoifexists{#1}%
  {%
    \bgroup
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \let\@org@glsnumberformat\glsnumberformat
    \def\glsnumberformat##1{##1}%
  }%
}

```

```

\protected@edef\the@numberlist{%
  \csname glo@\glo@label @numberlist\endcsname}%
\def\@gls@numlist@sep{%
\def\@gls@numlist@nextsep{%
\def\@gls@numlist@lastsep{%
\def\@gls@thislist{%
\def\@gls@donext@def{%
\renewcommand\do[1]{%
  \protected@edef\@gls@thislist{%
    \@gls@thislist
    \noexpand\@gls@numlist@sep
    ##1%
  }%
  \let\@gls@numlist@sep\@gls@numlist@nextsep
  \def\@gls@numlist@nextsep{\glsnumlistsep}%
  \@gls@donext@def
  \def\@gls@donext@def{%
    \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
  }%
}%
\expandafter \glsnumlistparser \expandafter{\the@numberlist}%
\let\@gls@numlist@sep\@gls@numlist@lastsep
\@gls@thislist
\egroup
}%
}
}
\newcommand*\glsnumlistsep{, }
\newcommand*\glsnumlistlastsep{ \& }
\newcommand*\gls hyperlink}[2][\glsentrytext{\glo@label}]{%
\def\glo@label{#2}%
\@glslink{\glo linkprefix\glsdetoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*\glsadd}[2][{}]{%
  \@gls@adjustmode
  \glsdoifexists{#2}%
  {%
    \def\@glsnumberformat{glsnumberformat}%
    \protected@edef\@gls@counter{\csname glo\glsdetoklabel{#2}@counter\endcsname}%
    \setkeys{glossadd}{#1}%
    \@gls@saveentrycounter
    \@gls@setsort{#2}%
    \@do@wrglossary{#2}%
  }%
}
}
\glsmfuexcl{\glsadd}
\newcommand*\@gls@adjustmode{}
\AtBeginDocument{\renewcommand*\@gls@adjustmode{\ifvmode\mbox{}\fi}}

```

```

\newrobustcmd*\glsaddall}[1][ ]{%
  \protected@edef\@glo@type{\@glo@types}%
  \setkeys{glossadd}{#1}%
  \forallglsentries[\@glo@type]{\@glo@entry}{%
    \glsadd[#1]{\@glo@entry}%
  }%
}
\newrobustcmd*\glsaddallunused}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glo@entry}%
  {%
    \ifglsused{\@glo@entry}{\@glsadd[format=glsignore]{\@glo@entry}}%
  }%
}
\newcommand*\glsignore}[1]{%
  \edef\glsopenbrace{\expandafter\@gobble\string\{ }
  \edef\glsclosebrace{\expandafter\@gobble\string\}}
  \edef\glsbackslash{\expandafter\@gobble\string\}
  \edef\glsquote#1{\string"#1\string"}
  \edef\glspercentchar{\expandafter\@gobble\string\%}
  \edef\glsstildechar{\string~}
  \ifglsxindy
    \newcommand*\@glsfirstletter}{A}
  \fi
  \newcommand*\GlsSetXdyFirstLetterAfterDigits){%
    \@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
  \ifglsxindy
    \newcommand*\@GlsSetXdyFirstLetterAfterDigits}[1]{%
      \renewcommand*\@glsfirstletter}{#1}}
    \newcommand*\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
      \renewcommand*\@glsfirstletter}{#1}%
      \@onelevel@sanitize\@glsfirstletter
    }
  \else
    \newcommand*\@GlsSetXdyFirstLetterAfterDigits}[1]{%
      \glsnoxywarning\GlsSetXdyFirstLetterAfterDigits}
    \newcommand*\s@GlsSetXdyFirstLetterAfterDigits){%
      \@GlsSetXdyFirstLetterAfterDigits
    }
  \fi
  \ifglsxindy
    \newcommand*\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
  \fi
  \newcommand*\GlsSetXdyNumberGroupOrder}[1]{%
    \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
  }
  \ifglsxindy
    \newcommand*\@GlsSetXdyNumberGroupOrder}[1]{%
      \renewcommand*\@xdynumbergrouporder}{#1}%
    }
  \newcommand*\s@GlsSetXdyNumberGroupOrder}[1]{%

```



```

\renewcommand*\@xdynumbergrouporder}{#1}%
\@onelevel@sanitize\@xdynumbergrouporder
}
\else
\newcommand*\@GlsSetXdyNumberGroupOrder}[1]{%
\glsnoxywarning\GlsSetXdyNumberGroupOrder}
\newcommand*\s@GlsSetXdyNumberGroupOrder}{%
\@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*\@glsminrange}{2}
\ifglsxindy
\newcommand*\@GlsSetXdyMinRangeLength}[1]{%
\renewcommand*\@glsminrange}{#1}}
\else
\newcommand*\@GlsSetXdyMinRangeLength}[1]{%
\glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglsxindy
\def\writeist{%
\ifundef\glswrite{\newwrite\glswrite}{}%
\@gls@addpredefinedattributes
\openout\glswrite=\istfilename
\write\glswrite{;; xindy style file created by the glossaries
package}%
\write\glswrite{;; for document '\jobname' on
\the\year-\the\month-\the\day}%
\write\glswrite{^^J; required styles^^J}
\@for\@xdystyle:=\@xdyrequiredstyles\do{%
\ifx\@xdystyle\@empty
\else
\protected@write\glswrite}{(require
\string"\@xdystyle.xdy\string")}%
\fi
}%
\write\glswrite{^^J%
; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%
\@for\@gls@classI:=\@gls@xdy@locationlist\do{%
\protected@write\glswrite}{(define-location-class
\string"\@gls@classI\string"^^J\space\space\space
(
:sep "{ }{"
\csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
:sep "}"
)
^^J\space\space\space

```

```

: min-range-length \@glsminrange^^J%
)
}%
{%
\@for\@gls@classII:=\@gls@xdy@locationlist\do{%
  \protected@write\glswrite{}\{(define-location-class
    \string"\@gls@classII-\@gls@classI\string"
    ^^J\space\space\space
    (
      :sep "{"
      \csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
      :sep "}\{"
      \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
      :sep "}"
    )
    ^^J\space\space\space
    : min-range-length \@glsminrange^^J%
  )
}%
}%
}%
}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\glsseeformat\string"
  :close \string"{}\string")}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
  :open \string"\string
  \glossarysection[\string\glossarytoctitle]{\string
  \glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
  {%
    \@for\@this@attr:=\@xdyattributelist\do{%
      \protected@write\glswrite{}\{\string\providecommand*%
        \expandafter\string
        \csname glsX\@this@ctr X\@this@attr\endcsname[2]%
        {%
          \string\setentrycounter
            [\expandafter\@gobble\string\#1]{\@this@ctr}%
          \expandafter\string

```

```

\csname\@this@attr\endcsname
  {\expandafter\@gobble\string\#2}%
}%
}%
}%
}%
\write\glswrite{%
  \string\begin
  {theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
  \space\space:close \string" \glpercentchar\glstildechar n\string
  \end{theglossary}\string\glossarypostamble
  \glstildechar n\string" ^^J\space\space\space
  :tree)}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \glstildechar n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list
  :sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
  :sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
  :open-head \string"\string\glsgroupheading
  \glsopenbrace\string"^^J\space\space\space
  :close-head \string"\glsclosebrace\string")}%

```

```

\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}%
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax
\ifundef{\glswrite}{\newwrite\glswrite}{}\relax
\openout\glswrite=\istfilename
\write\glswrite{\glspercentchar\space makeindex style file
created by the glossaries package}
\write\glswrite{\glspercentchar\space for document
'\jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\@gls@actualchar'}
\write\glswrite{encap '\@gls@encapchar'}
\write\glswrite{level '\@gls@levelchar'}
\write\glswrite{quote '\@gls@quotechar'}
\write\glswrite{keyword \string"string\glossaryentry\string"}
\write\glswrite{preamble \string"string\glossarysection[\string
\glossarytoctitle]{\string\glossarytitle}\string
\glossarypreamble\string\n\string\begin{theglossary}\string
\glossaryheader\string\n\string"}
\write\glswrite{postamble \string"string%\string\n\string
\end{theglossary}\string\glossarypostamble\string\n
\string"}
\write\glswrite{group_skip \string"string\glsgroupskip\string\n
\string"}
\write\glswrite{item_0 \string"string%\string\n\string"}
\write\glswrite{item_1 \string"string%\string\n\string"}
\write\glswrite{item_2 \string"string%\string\n\string"}
\write\glswrite{item_01 \string"string%\string\n\string"}
\write\glswrite{item_x1
\string"string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{item_12 \string"string%\string\n\string"}
\write\glswrite{item_x2
\string"string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{delim_0 \string"string{\string
\glossaryentrynumbers\string{\string\relax \string"}
\write\glswrite{delim_1 \string"string{\string

```

```

    \glossaryentrynumbers\string\{\string\relax \string}
\write\glswrite{delim_2 \string"\string\{\string
    \glossaryentrynumbers\string\{\string\relax \string}
\write\glswrite{delim_t \string"\string\}\string}\string}
\write\glswrite{delim_n \string"\string\delimN \string}
\write\glswrite{delim_r \string"\string\delimR \string}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
    \string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
    \string"\string}\string\relax
    \string\glsresetentrylist \string}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"glscpositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
    \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
    \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}
\fi
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{
\ifglxsindy
\newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\newcommand*\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}%
\@ifpackageloaded{tracklang}%
{
\IfTrackedLanguage{german}%
{
\def\@gls@extramakeindexopts{-g}%
}%
}%
}%
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkquote####1#####2#1###3\noexpand\null{

```

```

\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={###1}%
\noexpand\ifx\noexpand\null###2\noexpand\null
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###2#1#1\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###2#1###3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkquote###1\noexpand\@nil
#1#1\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkescquote###1\noexpand\@nil
\expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkescactual###1\noexpand\@nil
\noexpand\?\noexpand\?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%

```

```

\noexpand\expandafter\noexpand\@gls@checkactual####1\noexpand\@nil
\noexpand?\noexpand?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkbar####1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkeschar####1\noexpand\@nil
\noexpand\|\noexpand\|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checklevel####1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkescllevel####1\noexpand\@nil
\noexpand\!\noexpand\!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesc####1\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkesquote####1%
\expandonce{\csname#1\endcsname}####2\expandonce{\csname#1\endcsname}%
####3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={####1}%
\noexpand\ifx\noexpand\null####2\noexpand\null
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkesquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar}%

```

```

        \noexpand\def\noexpand\@gls@checkescquote{%
          \noexpand\@gls@checkescquote####3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
  \noexpand\@gls@quotechar\noexpand\string
  \expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\def\noexpand\@gls@checkescquote{%
  \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
  \expandonce{\csname#1\endcsname}\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkescquote{%
  \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
  ####3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkescquote
}%
}%
\@gls@docheckquotedef
}
\newcommand*\@gls@nosetquote}[1]{\PackageError{glossaries}%
  {\string\GlsSetQuote\space not permitted here}%
  {Move \string\GlsSetQuote\space earlier in the preamble, as
  soon as possible after glossaries.sty has been loaded}}
\fi
\newcommand*\@gls@extramakeindexopts}[1]{
\newcommand{\noist}{%
  \@gls@addpredefinedattributes
  \let\writeist\relax
}
\newcommand{\@gls@noist@toolate}{%
  \GlossariesWarning{too late to use \string\noist. The style file
  has already been written}%
}
\newcommand*\@makeglossary}[1]{%
  \ifglossaryexists{#1}%
  {%
    \ifglssavewrites
      \expandafter\newtoks\csname glo@#1@filetok\endcsname
    \else
      \expandafter\newwrite\csname glo@#1@file\endcsname
      \expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
    \fi
    \@gls@renewglossary
    \writeist
  }%
  {%

```



```

\PackageError{glossaries}%
{Glossary type ‘#1’ not defined}%
{New glossaries must be defined before using \string\makeglossaries}%
}%
}
\newcommand*{\@glsopenfile}[2]{%
\immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
\PackageInfo{glossaries}{Writing glossary file
\jobname.\csname @glotype@#2@out\endcsname}%
}
\newcommand*{\@closegls}[1]{%
\closeout\csname glo@#1@file\endcsname
}
\ifglsxindy
\newcommand*{\@gls@automake}[1]{%
\ifglossaryexists{#1}
{%
\@closegls{#1}%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-M ord/letorder }}%
{\let\@gls@order\@empty}%
\gls@xdy@set@lang{#1}%
\ifdefvoid\gls@codepage
{\let\@gls@codepage\@empty}%
{\def\@gls@codepage{-C \gls@codepage\space}}%
\edef\@gls@dothiswrite{\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \gls@istfilebase\space
\@gls@codepage
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}}%
}%
\@gls@dothiswrite
}%
}%
}
\else
\newcommand*{\@gls@automake}[1]{%
\ifglossaryexists{#1}
{%
\@closegls{#1}%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-l }}%
{\let\@gls@order\@empty}%
\edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order

```

```

        -s \istfilename\space
        -t \jobname.\csuse{@glotype@#1@log}
        -o \jobname.\csuse{@glotype@#1@in}
        \jobname.\csuse{@glotype@#1@out}}%
    }%
    \@gls@dothiswrite
}
{
    \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}
}
\fi
\ifglsxindy
\newcommand*{\@gls@automake@immediate}[1]{%
    \ifglossaryexists{#1}
    {%
        \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
        {%
            \ifdefstring{\glsorder}{letter}%
            {\def\@gls@order{-M ord/letorder }}%
            {\let\@gls@order\@empty}%
            \gls@xdy@set@lang{#1}%
            \ifdefvoid\gls@codepage
            {\let\@gls@codepage\@empty}%
            {\def\@gls@codepage{-C \gls@codepage\space}}%
            \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
                -I xindy
                \@gls@order
                -L \@gls@langmod\space
                -M \gls@istfilebase\space
                \@gls@codepage
                -t \jobname.\csuse{@glotype@#1@log}
                -o \jobname.\csuse{@glotype@#1@in}
                \jobname.\csuse{@glotype@#1@out}}%
            }%
            \@gls@dothiswrite
        }%
        {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
            doesn't exist. Rerun may be required}}%
    }%
    {
        \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
    }%
}
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
    \ifglossaryexists{#1}
    {%
        \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
        {%

```

```

\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-1 }}%
{\let\@gls@order\@empty}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order
-s \istfilename\space
-t \jobname.\csuse{@gloftype@#1@log}
-o \jobname.\csuse{@gloftype@#1@in}
\jobname.\csuse{@gloftype@#1@out}}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@gloftype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{%
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\fi
\newcommand*{\@warn@nomakeglossaries}{}
\newcommand*{\warn@nomakeglossaries}{\@warn@nomakeglossaries}
\newcommand{\@gls@automake@types}{\@glo@types}
\newcommand{\@gls@@automake@immediate}{%
\ifnum\gls@automake@nr=2\relax
\protected@edef\gls@tmp{\@gls@automake@types}%
\@for\@gls@type:=\gls@tmp\do{%
\ifdefempty{\@gls@type}{}%
{\@gls@automake@immediate{\@gls@type}}%
}%
\glsautomakefalse
\renewcommand*{\@gls@doautomake}{}%
\else
\ifdefempty\gls@automake@makegloss{}%
{%
\IfFileExists{\jobname.aux}%
{\immediate\write18{\gls@automake@makegloss\space \jobname}}{}%
\glsautomakefalse
\renewcommand*{\@gls@doautomake}{}%
}%
\fi
}
\newcommand*{\makeglossaries}{%
\ifglxsindy
\def\glsindexingsetting{xindy}%
\else
\def\glsindexingsetting{makeindex}%
\fi
\@domakeglossaries
{%
\@gls@@automake@immediate

```

```

\ifundef{\glswrite}{\newwrite\glswrite}{}%
\protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{}}
\protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{}}
\ifundef\@gls@extramakeindexopts
{%
}%
\protected@write\@auxout{}{\string\providecommand
\string\@gls@extramakeindexopts[1]{}}
\protected@write\@auxout{}{\string\@gls@extramakeindexopts
{\@gls@extramakeindexopts}}%
}%
\protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
\protected@write\@auxout{}{\string\@glsorder{\glsorder}}
\@for\@glo@type:=\@glo@types\do{%
\ifthenelse{equal{\@glo@type}{}}{}{%
\@makeglossary{\@glo@type}}%
}%
\renewcommand*\newglossary[4][]{%
\PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%
\let\@makeglossary\@gobble
\let\makeglossaries\relax
\@disable@onlypremakeg
\let\gls@checkseeallowed\relax
\let\warn@nomakeglossaries\relax
\let\warn@noprntglossary\@makeglossaries@warn@noprntglossary
\ifglssavenumberlist
\edef\@gls@dodolistparser{\noexpand\DeclareListParser
{\noexpand\glsnumlistparser}{\delimN}}%
\@gls@dodolistparser
\fi
\let\makenoidxglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
\let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
\ifglsautomake
\renewcommand*{\@gls@doautomake}{%
\protected@edef\gls@tmp{\@gls@automake@types}%
\@for\@gls@type:=\gls@tmp\do{%
\ifdefempty{\@gls@type}{}%
{\@gls@automake{\@gls@type}}%
}%
}%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}

```

```

\newcommand\@makeglossaries@warn@noprintglossary{%
  \ifdefstring{\@glo@types}{,}%
  {%
    \GlossariesWarningNoLine{No glossaries have been defined}%
  }%
  {%
    \GlossariesWarningNoLine{No \string\printglossary\space
      or \string\printglossaries\space
      found. ^^J(Remove \string\makeglossaries\space if you
      don't want any glossaries.) ^^JThis document will not
      have a glossary}%
  }%
}%
\AtEndDocument{%
  \warn@nomakeglossaries
  \warn@noprintglossary
}
\newcommand*\makenoidxglossaries{%
  \def\glsindexingsetting{noidx}%
  \@domakeglossaries
  {%
    \renewcommand{\@gls@noref@warn}[1]{%
      \GlossariesWarning{Empty glossary for
        \string\printnoidxglossary[type={##1}].
        Rerun may be required (or you may have forgotten to use
        commands like \string\gls)}%
    }%
    \let\@gls@checkmkidxchars\@gobble
    \glsesclocationsfalse
    \let\@do@wrglossary\gls@noidxglossary
    \let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
    \let\@gls@checkseeallowed\relax
    \renewcommand{\@do@seeglossary}[2]{%
      \protected@edef\@gls@label{\glsdetoklabel{##1}}%
      \protected@write\@auxout{}{%
        \string\@gls@reference
          {\csname glo@\@gls@label @type\endcsname}%
          {\@gls@label}%
          {%
            \string\glsseeformat##2}%
          }%
      }%
    }%
  }%
\AtBeginDocument
{%
  \write\@auxout{\string\providecommand\string\@gls@reference[3]{}%
}%
\def\warn@noprintglossary{%
  \GlossariesWarningNoLine{No \string\printnoidxglossary\space
    or \string\printnoidxglossaries ^^J

```

```

    found. (Remove \string\makenoidxglossaries\space if you
    don't want any glossaries.)^^JThis document will not have a glossary}%
}%
\let\warn@nomakeglossaries\relax
\let\makeglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
  \let\@glo@assign@sortkey\@glo@assign@sortkey
  \def\@glo@sorttype{\@glo@default@sorttype}%
}%
\renewcommand*\new@glossaryentry[2]{%
  \PackageError{glossaries}{Glossary entries must be
  defined in the preamble^^Jwhen you use
  \string\makenoidxglossaries}%
  {Either move your definitions to the preamble or use
  \string\makeglossaries}%
}%
\renewcommand*{\glsentrynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \glsnoidxloclist{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for '##1'. Either
      a rerun is required or you haven't referenced the entry}%
    }%
  }%
}%
\renewcommand*{\glsdisplaynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \def\@gls@noidxloclist@sep{%
      \def\@gls@noidxloclist@sep{%
        \def\@gls@noidxloclist@sep{%
          \glsnumlistsep
        }%
      }%
    }%
    \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
  }%
  \def\@gls@noidxloclist@finalsep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
  \@gls@noidxloclist@finalsep
  \@gls@noidxloclist@prev
}%
{%
  ??\glsdoifexists{##1}%

```

```

    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
}
\renewcommand*{\glsnumberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo\@glsdetoklabel{##1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc@glsnoidxdisplayloc
  \let\@gls@org@glsseeformat@glsseeformat
  \let\glsnoidxdisplayloc##2\relax
  \let\glsseeformat##3\relax
  \ifdef\@gls@loclist
  {%
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
  \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
  \let\glsseeformat\@gls@org@glsseeformat
}
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
\@glo@check@sortallowed\makenoidxglossaries
\let\@gls@savenonumberlist\gls@noidx@nonumberlist
}%
}
\@onlypreamble{\makenoidxglossaries}
\newcommand*{\glsnumberlistloop}[3]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
\newcommand*{\glsnoidxnumberlistloophandler}[1]{%
  #1%
}
\newcommand*{\@no@makeglossaries}{%
  \PackageError{glossaries}{You can’t use both
  \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
  together.}%
}
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
    is required to make \string\printnoidxglossary[type={##1}] work}%
}

```

```

}
\newcommand*\gls@noidxglossary{%
  \protected@write\auxout{\gls@wrglossdisablelocationcmds}{%
    \string\gls@reference
      {\csname glo@\gls@label @type\endcsname}%
      {\gls@label}%
      {\string\glsnoidxdisplayloc
        {\glo@counterprefix}%
        {\gls@counter}%
        {\glsnumberformat}%
        {\gls@locref}}%
      }%
  }%
}
\AtEndDocument{%
  \glswritefiles
}
\newcommand*\gls@writefiles{%
  \forallglossaries{\glo@type}{%
    \ifcsundef{glo@\glo@type @filetok}%
      {%
        \def\gls@tmp{}}%
      }%
      {%
        \edef\gls@tmp{\expandafter\the
          \csname glo@\glo@type @filetok\endcsname}%
        }%
      \ifx\gls@tmp\@empty
        \ifx\glo@type\glsdefaulttype
          \GlossariesWarningNoLine{Glossary ‘\glo@type’ has no
            entries.^^JRemember to use package option ‘nomain’ if
you
              don’t want to^^Juse the main glossary}%
          \else
            \GlossariesWarningNoLine{Glossary ‘\glo@type’ has no
              entries}%
          \fi
        \else
          \glsopenfile{\glswrite}{\glo@type}%
          \immediate\write\glswrite{%
            \expandafter\the
              \csname glo@\glo@type @filetok\endcsname}%
          \immediate\closeout\glswrite
        \fi
      }%
    }
}
\newcommand*\gls@glossary[1]{%
  \gls@glossary{#1}%
}
\newcommand*\gls@glossary[2]{%

```



```

\if@gls@debug
  \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
\fi
}
\newcommand{\@gls@renewglossary}{%
  \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
  \let\@gls@renewglossary\@empty
}
\newcommand*\@gls@wrglossary}[2]{%
  \ifglssavewrites
    \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
    \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
    \expandafter{\@gls@tmp~J}%
  \else
    \ifcsdef{glo@#1@file}%
      {%
        \expandafter\protected@write\csname glo@#1@file\endcsname{%
          \gls@disablepagerefexpansion}{#2}%
        }%
      {%
        \ifignoredglossary{#1}{}%
          {%
            \GlossariesWarning{No file defined for glossary ‘#1’}%
          }%
        }%
      }%
    \fi
  \endgroup\@esphack
}
\newcommand*\@do@wrglossary}[1]{%
  \gls@ifnotmeasuring{\gls@writeentry{#1}{\@do@wrglossary{#1}}}%
}
\newcommand*\@gls@writeentry}[2]{%
  \ifglsindexonlyfirst
    \ifglsused{#1}{#2}%
  \else
    #2%
  \fi
}
}
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
\newcommand*\@gls@disablepagerefexpansion{%
  \gls@wrgloss@disablelocationcmds
  \ifgls@esc@locations
    \@for\@gls@this:=\gls@protected@pagefmts\do
      {%
        \expandafter\let\@gls@this\relax
      }%
  \fi
}
}
\newcommand{\gls@wrgloss@disablelocationcmds}{}

```

```

\newcommand*\gls@alphpage}{\@alph\c@page}
\newcommand*\gls@Alphpage}{\@Alph\c@page}
\newcommand*\gls@numberpage}{\number\c@page}
\newcommand*\gls@arabicpage}{\@arabic\c@page}
\newcommand*\gls@romanpage}{\romannumeral\c@page}
\newcommand*\gls@Romanpage}{\@Roman\c@page}
\newcommand*\glsaddprotectedpagefmt}[1]{%
  \ifglsclocations
    \protected@eappto\gls@protected@pagefmts{\expandonce{\csname gls#1page\endcsname}}%
    \csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
    \protected@eappto\@wrglossarynumberhook{%
      \noexpand\let\expandonce{\csname org@gls#1\endcsname}%
      \expandonce{\csname#1\endcsname}%
      \noexpand\def\expandonce{\csname#1\endcsname}{%
        \noexpand\@wrglossary@pageformat
          \expandonce{\csname gls#1page\endcsname}%
          \expandonce{\csname org@gls#1\endcsname}%
        }%
      }%
    }%
  \else
    \GlsWarnAddProtectedPageFmt
  \fi
}
\newcommand*\GlsWarnAddProtectedPageFmt){%
  \GlossariesWarning{\string\glsaddprotectedpagefmt\space has no
    effect with esclocations=false}%
}
\newcommand*\@wrglossarynumberhook{}
\newcommand*\@wrglossary@pageformat}[3]{%
  \ifx#3\c@page #1\else #2#3\fi
}
\newcommand*\@do@wrglossary}[1]{%
  \ifglsclocations
    \@do@esc@wrglossary{#1}%
  \else
    \@do@noesc@wrglossary{#1}%
  \fi
}
\newcommand*\glswrglossdisableanchorcmds}{\let\glstexorpdfstring\@secondoftwo}
\ifdef\texorpdfstring
  {\appto\glswrglossdisableanchorcmds{\let\texorpdfstring\@secondoftwo}}{}
\ifdef\pdfstringdefPreHook
  {\appto\glswrglossdisableanchorcmds{\pdfstringdefPreHook}}{}
\newcommand*\glswrglosslocationtarget}[1]{#1}
\newcommand*\glswrglosslocationtextfmt}[1]{#1}
\newcommand*\@do@noesc@wrglossary}[1]{%
  \expandafter\def\expandafter\@glslocref\expandafter{\theglentrycounter}%
  \expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
  \ifx\@glsHlocref\@glslocref
    \def\@glo@counterprefix{}%
  \fi
}

```

```

\else
  \xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
\fi
\protected@edef\@gls@label{\glsdetoklabel{#1}}%
\@do@wrglossary
}
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodsfalse
\newcommand*\@do@esc@wrglossary}[1]{% please read documented code!
\begingroup
  \let\gls@orgthe\the
  \let\gls@orgnumber\number
  \let\gls@orgarabic\@arabic
  \let\gls@orgromannumeral\romannumeral
  \let\gls@orgalph\@alph
  \let\gls@orgAlph\@Alph
  \let\gls@orgRoman\@Roman
  \ifglswrallowprimitivemods
    \def\gls@the##1{%
      \ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
    \def\the{\expandafter\gls@the}%
    \def\gls@number##1{%
      \ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
    \def\number{\expandafter\gls@number}%
    \def\romannumeral##1{%
      \ifx##1\c@page \gls@romanpage \else \gls@orgromannumeral##1\fi}%
  \fi
  \def\@arabic##1{%
    \ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
  \def\@Roman##1{%
    \ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
  \def\@alph##1{%
    \ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
  \def\@Alph##1{%
    \ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
  \@wrglossarynumberhook
  \gls@disablepagerefexpansion
  \protected@xdef\@glslocref{\theHglentrycounter}%
\endgroup
\@gls@checkmkidxchars\@glslocref
\expandafter\ifx\theHglentrycounter\theHglentrycounter\relax
  \def\@glo@counterprefix{}%
\else
  \protected@edef\@glsHlocref{\theHglentrycounter}%
  \@gls@checkmkidxchars\@glsHlocref
  \xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
\fi
\protected@edef\@gls@label{\glsdetoklabel{#1}}%
\@do@wrglossary
}

```

```

\newcommand*{\@do@wrglossary}{%
  \ifglxindy
    \expandafter\glo@check@mkidxrangear\@glsnumberformat\@nil
    \def\@glo@range{}%
    \expandafter\if\@glo@prefix(\relax
      \def\@glo@range{:open-range}%
    \else
      \expandafter\if\@glo@prefix)\relax
      \def\@glo@range{:close-range}%
    \fi
    \fi
    \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
      (indexentry :tkey (\csname glo@\@gls@label @index\endcsname)
        :locref \string{\@glo@counterprefix}{\@glslocref}\string" %
        :attr \string"\@gls@counter\@glo@suffix\string"
        \@glo@range
      )
    }%
  \else
    \set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@glsnumberformat}%
    {\@glo@counterprefix}%
    \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
      \string@glossaryentry{\csname glo@\@gls@label @index\endcsname
        \@gls@encapchar\@glo@numfmt}{\@glslocref}}%
    \fi
  }
\newcommand*\@xp@gls@getcounterprefix[2]{%
  \bgroup
  \glswrglossdisableanchorcmds
  \protected@edef\@do@gls@getcounterprefix{%
    \noexpand\egroup
    \noexpand\@gls@getcounterprefix{#1}{#2}%
  }%
  \@do@gls@getcounterprefix
}
\newcommand*\@gls@getcounterprefix[2]{%
  \def\@gls@thisloc{#1}\def\@gls@thisHloc{#2}%
  \ifx\@gls@thisloc\@gls@thisHloc
    \def\@glo@counterprefix{}%
  \else
    \def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
      \def\@glo@tmp{##2}%
      \ifx\@glo@tmp\@empty
        \def\@glo@counterprefix{}%
      \else
        \def\@glo@counterprefix{##1}%
      \fi
    }%
    \@gls@get@counterprefix#2.#1\end@getprefix
    \ifx\@glo@counterprefix\@empty

```

```

\GlossariesWarning{Hyper target ‘#2’ can’t be formed by
  prefixing location ‘#1’. You need to modify the
  definition of \string\theH\@gls@counter otherwise you
  will get the warning: “name{\@gls@counter.#1}’ has been
  referenced but does not exist”}%
\fi
\fi
}
\newcommand{\@do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref
\ifglsxindy
\gls@glossary{\csname glo@#1@type\endcsname}{%
(indexentry
:key (\csname glo@#1@index\endcsname)
:xref (\string\@gls@xref\string)
:attr \string"see\string"
)
}%
\else
\gls@glossary{\csname glo@#1@type\endcsname}{%
\string\glossaryentry{\csname glo@#1@index\endcsname
\@gls@encapchar glsseeformat\@gls@xref}{Z}}%
\fi
}
\def\@gls@fixbraces#1#2#3\@nil{%
\ifx#2[\relax
\@gls@fixbraces#1#2#3\@end@fixbraces
\else
\def#1{{#2#3}}%
\fi
}
\def\@gls@fixbraces#1[#2]#3\@end@fixbraces{%
\def#1{[#2]{#3}}%
}
\newrobustcmd*\glssee}[3][\seename]{%
\@do@seeglossary{#2}{[#1]{#3}}}
\newcommand*\@glssee}[3][\seename]{%
\glssee[#1]{#3}{#2}}
\newrobustcmd*\glsseeformat}[3][\seename]{%
\emph{#1} \glsseelist{#2}}
\newrobustcmd*\glsseelist}[1]{%
\let\@gls@dolast\relax
\let\@gls@donext\relax
\@for\@gls@thislabel:=#1\do{%
\ifx\@xfor@nextelement\@nnil
\@gls@dolast
\else
\@gls@donext

```

```

\fi
\expandafter\glsseeitem\expandafter{\@gls@thislabel}%
\let\@gls@dolast\glsseelastsep
\let\@gls@donext\glsseesep
}%
}
\newcommand*\glsseelastsep{\space\andname\space}
\newcommand*\glsseesep{, }
\newrobustcmd*\glsseeitem[1]{\gls hyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*\glsseeitemformat[1]{%
\ifgls hasshort{#1}{\glsentrytext{#1}}{\glsentryname{#1}}%
}
\newcommand*\gls@save@numberlist[1]{%
\ifgls savenumberlist
\toks@{#1}%
\protected@edef\@do@writeaux@info{%
\noexpand\csgdef{glo@glscurrententrylabel @numberlist}{\the\toks@}%
}%
\@onelevel@sanitize\@do@writeaux@info
\protected@write\@auxout{}{\@do@writeaux@info}%
\fi
}
\newcommand*\warn@noprintglossary{}%
\ifcsundef{printglossary}{}%
{%
\@gls@warnonglossdefined
\undef\printglossary
}
\newcommand*\@printgloss@checkexists@noignored}[2]{%
\ifglossaryexists{#1}%
{#2}%
{%
\ifignoredglossary{#1}%
{\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
{\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}%
}
\newcommand*\@printgloss@checkexists@allowignored}[2]{%
\s@ifglossaryexists{#1}%
{#2}%
{\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\newcommand*\printglossary[1][type=\glsdefaulttype]{%
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\@printglossary{#1}{\@print@glossary}%
}
\newcommand*\printglossaries{%
\forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}

```

```

\newcommand*\printnoidxglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@noidx@glossary}%
}
\newcommand*\printnoidxglossaries{%
  \foralllglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
\newcommand*\@printgloss@setsort{}
\newcommand*\@gls@preglossaryhook{}
\newcommand{\@printglossary}[2]{%
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{\csname @glo@type@\@glo@type @title\endcsname}%
  \def\glossarytoctitle{\glossarytitle}%
  \let\org@glossarytitle\glossarytitle
  \def\@glossarystyle{%
    \ifx\@glossary@default@style\relax
      \GlossariesWarning{No default glossary style provided \MessageBreak
        for the glossary ‘\@glo@type’. \MessageBreak
        Using fallback. \MessageBreak
        To fix this set the style with \MessageBreak
        \string\setglossarystyle\space or use the \MessageBreak
        style key=value option}%
    \fi
  }%
  \def\gls@dotoc@title{\glssettoctitle{\@glo@type}}%
  \let\org@glossaryentrynumbers\glossaryentrynumbers
  \gls@printglossary@begin
  \@printgloss@setsort
  \setkeys{printgloss}{#1}%
  \@printgloss@checkexists{\@glo@type}%
  {%
    \ifx\glossarytitle\org@glossarytitle
    \else
      \expandafter\let\csname @glo@type@\@glo@type @title\endcsname
        \glossarytitle
    \fi
    \let\currentglossary\@glo@type
    \let\org@glossaryentrynumbers\glossaryentrynumbers
    \let\glsnonextpages\@glsnonextpages
    \let\glsnextpages\@glsnextpages
    \let\nopostdesc\@nopostdesc
    \gls@dotoc@title
    \@glossarystyle
    \let\gls@org@glossaryentryfield\glossentry
    \let\gls@org@glossarysubentryfield\subglossentry
    \renewcommand{\glossentry}[1]{%
      \protected@xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
      \gls@org@glossaryentryfield{##1}%
    }%
    \renewcommand{\subglossentry}[2]{%

```

```

        \protected@xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
        \gls@org@glossarysubentryfield{##1}{##2}%
    }%
    \@gls@preglossaryhook
    #2%
    }%
\gls@printglossary@end
\global\let\glossaryentrynumbers\org@glossaryentrynumbers
\global\let\warn@noprntglossary\relax
}
\newcommand{\gls@printglossary@begin}{\begingroup}
\newcommand{\gls@printglossary@end}{\endgroup}
\expandafter\endgroup\if@endpe\@doendpe\fi
}
\newcommand{\@print@glossary}{%
    \makeatletter
    \input@{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%
    \IfFileExists{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%
    {}%
    {\null}%
    \ifglsxindy
        \gls@xdy@set@lang{\@glo@type}%
        \edef\@do@auxoutstuff{%
            \noexpand\AtEndDocument{%
                \noexpand\immediate\noexpand\write\@auxout{%
                    \string\providecommand\string\@xdylanguage[2]{}%
                }%
                \noexpand\immediate\noexpand\write\@auxout{%
                    \string\@xdylanguage{\@glo@type}{\@gls@langmod}%
                }%
            }%
        }%
        \do@auxoutstuff
        \edef\@do@auxoutstuff{%
            \noexpand\AtEndDocument{%
                \noexpand\immediate\noexpand\write\@auxout{%
                    \string\providecommand\string\@gls@codepage[2]{}%
                }%
                \noexpand\immediate\noexpand\write\@auxout{%
                    \string\@gls@codepage{\@glo@type}{\@gls@codepage}%
                }%
            }%
        }%
        \do@auxoutstuff
    \fi
\renewcommand*{\@warn@nomakeglossaries}{%
    \GlossariesWarningNoLine{\string\makeglossaries\space
    hasn't been used,^^Jthe glossaries will not be updated}%
}%
}
\newcommand*{\@glo@sortentries}[2]{%
    \glosortentrieswarning
    \def\@glo@sortinglist{}%
    \def\@glo@sortinghandler{#1}%

```



```

\protected@edef\@glo@type{#2}%
\forlistcsloop{\@glo@do@sortentries}{@glsref@#2}%
\csdef{@glsref@#2}{}%
\@for\@this@label:=\@glo@sortinglist\do{%
  \xifinlistcs{\@this@label}{@glsref@#2}%
  {%
    \listcsxadd{@glsref@#2}{\@this@label}%
  }%
  \ifcsdef{@glo@sortingchildren@\@this@label}%
  {%
    \@glo@addchildren{#2}{\@this@label}%
  }%
  {%
  }%
}
}
\newcommand*\@glo@addchildren}[2]{%
  \bgroup
  \letcs{\@glo@childlist}{@glo@sortingchildren@#2}%
  \@for\@this@childlabel:=\@glo@childlist\do
  {%
    \xifinlistcs{\@this@childlabel}{@glsref@#1}%
    {%
      \listcsxadd{@glsref@#1}{\@this@childlabel}%
    }%
    \ifcsdef{@glo@sortingchildren@\@this@childlabel}%
    {%
      \@glo@addchildren{#1}{\@this@childlabel}%
    }%
    {%
    }%
  }%
  \egroup
}
\newcommand*\@glo@do@sortentries}[1]{%
  \ifglshasparent{#1}%
  {%
    \protected@edef\@glo@parent{\csuse{glo@glsdetoklabel{#1}@parent}}%
    \ifcsundef{@glo@sortingchildren@\@glo@parent}%
    {%
      \csdef{@glo@sortingchildren@\@glo@parent}{}%
    }%
    {%
    }%
    \expandafter\@glo@sortedinsert
    \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
    \xifinlistcs{\@glo@parent}{@glsref@\@glo@type}%
    {%
    }%
    {%
  }
}

```

```

        \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
    }%
}%
{%
    \glo@sortedinsert{\@glo@sortinglist}{#1}%
}%
}
\newcommand*\@glo@sortedinsert}[2]{%
    \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
}%
\newcommand*\@glo@sorthandler@word}[2]{%
    \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef\glo@do@compare{%
        \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}
\newcommand*\@glo@sorthandler@letter}[2]{%
    \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef\glo@do@compare{%
        \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}
\newcommand*\@glo@sorthandler@case}[2]{%
    \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef\glo@do@compare{%
        \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}
\newcommand*\@glo@sorthandler@nocase}[2]{%
    \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef\glo@do@compare{%
        \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}
}

```

```

\newcommand*{\@glo@sortmacro@word}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@word}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=word]}{}}%
  }%
}
\newcommand*{\@glo@sortmacro@letter}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=letter]}{}}%
  }%
}
\newcommand*{\@glo@sortmacro@standard}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \ifcsdef{\@glo@sorthandler@glorder}%
    {%
      \@glo@sortentries{\cuse{\@glo@sorthandler@glorder}}{#1}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\glorder’}{}}%
    }%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=standard]}{}}%
  }%
}
\newcommand*{\@glo@sortmacro@case}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@case}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=case]}{}}%
  }%
}

```

```

\newcommand*{\@glo@sortmacro@nocase}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=nocase]}{}}%
  }%
}
\newcommand*{\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{%
    \forlsglentries[#1]{\@gls@thislabel}%
    {%
      \xifinlistcs{\@gls@thislabel}{\@glsref@#1}%
      {%
        \listead{\@glo@sortinglist}{\@gls@thislabel}%
      }%
      {%
        }%
      }%
    }%
    \cslet{\@glsref@#1}{\@glo@sortinglist}%
  }
}
\newcommand*{\@glo@sortmacro@def@do}[1]{%
  \ifinlistcs{#1}{\@glsref@\@glo@type}%
  {}%
  {%
    \listcsadd{\@glsref@\@glo@type}{#1}%
  }%
  \ifcsdef{\@glo@sortingchildren@#1}%
  {%
    \@glo@addchildren{\@glo@type}{#1}%
  }%
  {}%
}
\newcommand*{\@glo@sortmacro@use}[1]{%
}
\newcommand*{\@print@noidx@glossary}{%
  \ifcsdef{\@glsref@\@glo@type}%
  {%
    \ifcsdef{\@glo@sortmacro@\@glo@sorttype}%
    {%
      \csuse{\@glo@sortmacro@\@glo@sorttype}{\@glo@type}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}}%
    }%
  }%
  \glossarysection[\glossarytoctitle]{\glossarytitle}%
  \glossary preamble
  \def\@gls@currentlettergroup{}%
}

```

```

    \begin{theglossary}%
    \glossaryheader
    \glsresetentrylist
    \forlistcsloop{\@gls@noidx@do}{\@gls@ref@{\@glo@type}}%
    \end{theglossary}%
    \glossarypostamble
  }%
  {%
    \@gls@noref@warn{\@glo@type}%
  }%
}
\def\glo@grabfirst#1#2\@nil{%
\def\@gls@firsttok{#1}%
\ifdefempty\@gls@firsttok
  {%
    \def\@glo@thislettergrp{0}%
  }%
  {%
    \@onelevel@sanitize\@gls@firsttok
    \expandafter\@glo@grabfirst\@gls@firsttok{}{}\@nil
  }%
}
\def\@glo@grabfirst#1#2\@nil{%
\ifdefempty\@glo@thislettergrp
  {%
    \def\@glo@thislettergrp{glssymbols}%
  }%
  {%
    \count@=\uccode‘#1\relax
    \ifnum\count@=0\relax
      \def\@glo@thislettergrp{glssymbols}%
    \else
      \ifdefstring\@glo@sorttype{case}%
        {%
          \count@=‘#1\relax
        }%
        {%
          }%
        }%
      \edef\@glo@thislettergrp{\the\count@}%
    \fi
  }%
}
\newcommand{\@gls@noidx@do}[1]{%
\global\letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
\ifglshasparent{#1}%
  {%
    \gls@level=\csuse{glo@\glsdetoklabel{#1}@level}\relax
    \ifdefvoid{\@gls@loclist}
      {%
        \subglossentry{\gls@level}{#1}{}%
      }%
    }%
  }%
}

```

```

}%
{%
  \subglossentry{\gls@level}{#1}%
  {%
    \glsnoidxprenumberlist{#1}%
    \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
  }%
}%
}%
{%
  \letcs{\@gls@sort}{glo@\glsdetoklabel{#1}@sort}%
  \expandafter\glo@grabfirst\@gls@sort{}{}\@nil
  \ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
  {}%
  {%
    \ifdefempty{\@gls@currentlettergroup}{}%
    {%
      \global\let\@glo@thislettergrp\@glo@thislettergrp
      \glsgroupskip
    }%
    \glsgroupheading{\@glo@thislettergrp}%
  }%
  \global\let\@gls@currentlettergroup\@glo@thislettergrp
  \ifdefvoid{\@gls@loclist}
  {%
    \glossentry{#1}{}%
  }%
  {%
    \glossentry{#1}%
    {%
      \glsnoidxprenumberlist{#1}%
      \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
    }%
  }%
}%
}
\newcommand*{\glsnoidxloclist}[1]{%
  \def\@gls@noidxloclist@sep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxloclisthandler}{#1}%
}
\newcommand*{\glsnoidxloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%
    \@gls@noidxloclist@sep
    #1%
    \def\@gls@noidxloclist@sep{\delimN}%
    \def\@gls@noidxloclist@prev{#1}%
  }%
}

```

```

    }%
}
\newcommand*\glsnoidxdisplayloclisthandler[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%
    \@gls@noidxloclist@sep
    \@gls@noidxloclist@prev
    \def\@gls@noidxloclist@prev{#1}%
  }%
}
\newcommand*\glsnoidxdisplayloc[4]{%
  \setentrycounter[#1]{#2}%
  \csuse{#3}{#4}%
}
\newcommand*\@gls@reference[3]{%
  \glsdoifexistsorwarn{#2}%
  {%
    \ifcsundef{\@glsref@#1}{\csgdef{\@glsref@#1}{}}{}%
    \ifinlistcs{#2}{\@glsref@#1}%
    {}%
    \listcsgadd{\@glsref@#1}{#2}%
    \ifcsundef{glo@\glsdetoklabel{#2}@loclist}%
    {\csgdef{glo@\glsdetoklabel{#2}@loclist}{}}%
    {}%
    \listcsgadd{glo@\glsdetoklabel{#2}@loclist}{#3}%
  }%
}
\define@key{printgloss}{type}{\def\@glo@type{#1}}
\define@key{printgloss}{title}{%
  \def\glossarytitle{#1}%
  \let\gls@dotoc@title\relax
}
\define@key{printgloss}{toctitle}{%
  \def\glossarytoctitle{#1}%
  \let\gls@dotoc@title\relax
}
\define@key{printgloss}{style}{%
  \ifcsundef{\@glsstyle@#1}%
  {%
    \PackageError{glossaries}%
    {Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \def\@glossarystyle{\setglossarystyle{#1}}%
  }%
}
\define@choicekey{printgloss}{numberedsection}%
[\@gls@numberedsection@val\@gls@numberedsection@nr]%

```

```

{false,nolabel,autolabel,nameref}[nolabel]%
{%
\ifcase\gls@numberedsection@nr\relax
\renewcommand*{\@@glossarysecstar}{*}%
\renewcommand*{\@@glossaryseclabel}{}%
\or
\renewcommand*{\@@glossarysecstar}{}%
\renewcommand*{\@@glossaryseclabel}{}%
\or
\renewcommand*{\@@glossarysecstar}{}%
\renewcommand*{\@@glossaryseclabel}{\label{\glsautoprefix\@glo@type}}%
\or
\renewcommand*{\@@glossarysecstar}{*}%
\renewcommand*{\@@glossaryseclabel}{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}}%
\fi
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
\csuse{glsnogroupskip#1}%
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
\csuse{glsnopostdot#1}%
}
\newcommand*{\GlsEntryCounterLabelPrefix}{glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
\csuse{glsentrycounter#1}%
\@gls@define@glossaryentrycounter
}
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
\csuse{glsesubentrycounter#1}%
\@gls@define@glossarysubentrycounter
}
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
\ifglsnonumberlist
\def\glossaryentrynumbers##1{%
\else
\def\glossaryentrynumbers##1{##1}%
\fi}
\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
\newcommand*{\@glo@no@assign@sortkey}[1]{%
\PackageError{glossaries}{‘sort’ key not permitted with
\string\printglossary}%
{The ‘sort’ key may only be used with \string\printnoidxglossary}%
}
\newcommand*{\@glo@assign@sortkey}[1]{%
\def\@glo@sorttype{#1}%
}
\newcommand*{\@glsnonextpages}{%
\gdef\glossaryentrynumbers##1{%

```



```

        \glsresetentrylist
    }%
}
\newcommand*{\@glsnextpages}{%
    \gdef\glossaryentrynumbers##1{%
        ##1\glsresetentrylist}}
\newcommand*{\glsresetentrylist}{%
    \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
\newcommand*{\glsnonextpages}{}
\newcommand*{\glsnextpages}{}
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
\newcommand*{\glsresetsubentrycounter}{%
    \ifglssubentrycounter
        \setcounter{glossarysubentry}{0}%
    \fi
}
\newcommand*{\glsresetentrycounter}{%
    \ifglentrycounter
        \setcounter{glossaryentry}{0}%
    \fi
}
\newcommand*{\glsstepentry}[1]{%
    \ifglentrycounter
        \refstepcounter{glossaryentry}%
        \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \fi
}
\newcommand*{\glsstepsubentry}[1]{%
    \ifglssubentrycounter
        \protected@edef\currentglssubentry{\glsdetoklabel{#1}}%
        \refstepcounter{glossarysubentry}%
        \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
    \fi
}
\newcommand*{\glsrefentry}[1]{%
    \ifglentrycounter
        \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
        \ifglssubentrycounter
            \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
        \else
            \gls{#1}%
        \fi
    \fi
}
\newcommand*{\glsentrycounterlabel}{%
    \ifglentrycounter
        \theglossaryentry.\space
    \fi
}

```

```

}
\newcommand*{\glssubentrycounterlabel}{%
\ifglssubentrycounter
\theglossarysubentry)\space
\fi
}
\newcommand*{\glssubentryitem}[1]{%
\ifglssubentrycounter
\glssubentry{#1}\glssubentrycounterlabel
\else
\glssubentry{#1}
\fi
}
\newcommand*{\glssubentryitem}[1]{%
\ifglssubentrycounter
\glssubentry{#1}\glssubentrycounterlabel
\fi
}
\ifcsundef{theglossary}%
{%
\newenvironment{theglossary}{}{}%
}%
{%
\@gls@warnontheGLOSSDEFINED
\renewenvironment{theglossary}{}{}%
}
\newcommand*{\glossaryheader}{}
\newcommand*{\glstarget}[2]{\@glstarget{\glo@linkprefix#1}{#2}}
\newcommand*{\glossentry}[2]{\glossentryname{#1} #2}
\newcommand*{\glossentryname}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\letcs{\glo@name}{\glo@GLSDetoklabel{#1}@name}%
\expandafter\glsnamefont\expandafter{\glo@name}%
}%
}
\newcommand*{\Glossentryname}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\glsnamefont{\Glsentryname{#1}}%
}%
}
\glsmfuaddmap{\glossentryname}{\Glossentryname}
\newcommand*{\glossentrydesc}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\glsentrydesc{#1}%
}%
}
\newcommand*{\Glossentrydesc}[1]{%

```

```

\glsdoifexistsorwarn{#1}%
{%
  \Glsentrydesc{#1}%
}%
}
\glsmfuaddmap{\glossentrydesc}{\Glossentrydesc}
\newcommand*{\glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrysymbol{#1}%
  }%
}
\newcommand*{\Glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrysymbol{#1}%
  }%
}
\glsmfuaddmap{\glossentrysymbol}{\Glossentrysymbol}
\newcommand{\subglossentry}[3]{\glossentryname{#2} #3}
\newcommand*{\glsgroupskip}{}
\newcommand*{\glsgroupheading}[1]{}
\newrobustcmd*{\glsgetgrouptitle}[1]{%
  \@gls@getgrouptitle{#1}{\@gls@grptitle}%
  \@gls@grptitle
}
\ExplSyntaxOn
\newcommand*{\@gls@getgrouptitle}[2]{%
  \def#2{#1}%
  \regex_match:nnT { ~\A(.|glsnumbers|glssymbols)\Z } { #1 }
  { \ifcsundef{#1groupname}{ }\letcs#2{#1groupname} }
}
\ExplSyntaxOff
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%
  \DTLifint{#1}%
  {\edef#2{\char#1\relax}}%
  {%
    \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
  }%
}
\ExplSyntaxOn
\newcommand*{\glsgetgrouplabel}[1]{%
  \str_if_eq:eeTF {#1} { \glssymbolsgroupname }
  { \glssymbols }
  {
    \str_if_eq:eeTF {#1} { \glsnumbersgroupname }
    { \glsnumbers }
    { #1 }
  }
}
}

```

```

\ExplSyntaxOff
\newcommand*{\setentrycounter}[2][{}]{%
  \def\@glo@counterprefix{#1}%
  \ifx\@glo@counterprefix\@empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1.}%
  \fi
  \def\glsentrycounter{#2}%
}
\newcommand*{\setglossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \csname @glsstyle@#1\endcsname
  }%
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\newcommand{\newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
\newcommand{\renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
\newcommand*{\glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\glsnumber#1{#1}%
}%
{%
  \def\glsnumber#1{\@xp@glshypernumber{#1}}
}
\newrobustcmd{\@xp@glshypernumber}[1]{%
  \@glshypernumber#1\nohyperpage{}\@nil
}

```

```

}
\def\@gls@hypernumber#1\nohyperpage#2#3\@nil{%
  \ifx\#1\%
  \else
    \gls@delimR#1\delimR\delimR\%
  \fi
  \ifx\#2\%
  \else
    #2%
  \fi
  \ifx\#3\%
  \else
    \@gls@hypernumber#3\@nil
  \fi
}
\def\gls@delimR#1\delimR #2\delimR #3\{%
  \ifx\#2\%
  \gls@delimN{#1}%
  \else
    \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
  \fi
}
\def\gls@delimN#1{\gls@delimN#1\delimN \delimN\}
\def\gls@delimN#1\delimN #2\delimN#3\{%
  \ifx\#3\%
  \gls@numberlink{#1}%
  \else
    \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
  \fi
}
\def\@gls@numberlink#1{%
  \begingroup
  \toks@={}%
  \@gls@removespaces#1 \@nil
  \endgroup
}
\def\@gls@removespaces#1 #2\@nil{%
  \toks@=\expandafter{\the\toks@#1}%
  \ifx\#2\%
  \edef\@gls@x{\the\toks@}%
  \ifx\@gls@x\empty
  \else
  \bgroup
  \gls@wrglossdisableanchorcmds
  \protected@edef\@gls@tmp{\gls@entrycounter\@glo@counterprefix
  \gls@wrglosslocationtarget{\@gls@x}}%
  \@onelevel@sanitize\@gls@tmp
  \edef\@gls@x{\noexpand\egroup
  \noexpand\hyperlink{\@gls@tmp}%
  {\noexpand\gls@wrglosslocationtextfmt{\the\toks@}}%

```

```

    }%
    \@gls@x
  \fi
\else
  \@gls@ReturnAfterFi{%
    \@gls@removespaces#2\@nil
  }%
\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
\newcommand*\hyperrm[1]{\textrm{\glsnumber{#1}}}
\newcommand*\hypersf[1]{\textsf{\glsnumber{#1}}}
\newcommand*\hypertt[1]{\texttt{\glsnumber{#1}}}
\newcommand*\hyperbf[1]{\textbf{\glsnumber{#1}}}
\newcommand*\hypermd[1]{\textmd{\glsnumber{#1}}}
\newcommand*\hyperit[1]{\textit{\glsnumber{#1}}}
\newcommand*\hypersl[1]{\textsl{\glsnumber{#1}}}
\newcommand*\hyperup[1]{\textup{\glsnumber{#1}}}
\newcommand*\hypersc[1]{\textsc{\glsnumber{#1}}}
\newcommand*\hyperemph[1]{\emph{\glsnumber{#1}}}
\newcommand\oldacronym[4][\gls@label]{%
  \def\gls@label{#2}%
  \newacronym[4]{#1}{#2}{#3}%
  \ifcsundef{xspace}%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}}%
  }%
}%
\ifcsundef{xspace}%
{%
  \expandafter\edef\csname#1\endcsname{%
    \noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
      \noexpand\gls{#1}\noexpand\xspace}%
  }%
}%
}
\newcommand\newacronym[4][{}]{%
  \gls@newacronym{#1}{#2}{#3}{#4}%
}
\newcommand\gls@newacronym[4]{%
  \renewcommand\gls@newacronym[4]{%
    \PackageError{glossaries}{incompatible acronym style:
      rollback may be required}%
    {something has gone wrong with \string\newacronym. Try
      \string\usepackage{glossaries}[=v4.49]}%
  }%
  \setacronymstyle{long-short}%
  \newacronym[4]{#1}{#2}{#3}{#4}%
}
\newcommand*\acrpluralsuffix{\glsacrpluralsuffix}

```

```

\newrobustcmd*{\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
\newcommand*{\glsshortkey}{short}
\newcommand*{\glsshortpluralkey}{shortplural}
\newcommand*{\glslongkey}{long}
\newcommand*{\glslongpluralkey}{longplural}
\newcommand{\acrlinkfullformat}[5]{%
  \acrfullformat{#1{#3}{#4}[#5]}{#2{#3}{#4}[]}%
}
\newcommand{\acrfullformat}[2]{#1\glsspace{#2}}
\newrobustcmd{\glsspace}{\space}
\newcommand*{\acrfullfmt}[3]{%
  \acrlinkfullformat{\@acrshort}{#1}{#2}{#3}%
}
\newcommand*{\Acrfullfmt}[3]{%
  \acrlinkfullformat{\@Acrshort}{#1}{#2}{#3}%
}
\newcommand*{\ACRfullfmt}[3]{%
  \acrlinkfullformat{\@ACRshort}{#1}{#2}{#3}%
}
\newcommand*{\acrfullplfmt}[3]{%
  \acrlinkfullformat{\@acrshortpl}{#1}{#2}{#3}%
}
\newcommand*{\Acrfullplfmt}[3]{%
  \acrlinkfullformat{\@Acrshortpl}{#1}{#2}{#3}%
}
\newcommand*{\ACRfullplfmt}[3]{%
  \acrlinkfullformat{\@ACRshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfull}{\@gls@hyp@opt\ns@acrfull}
\newcommand*\ns@acrfull[2][]{%
  \new@ifnextchar[{\@acrfull{#1}{#2}}%
    {\@acrfull{#1}{#2}[]}%
}
\def\@acrfull#1#2[#3]{%
  \acrfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\Acrfull}{\@gls@hyp@opt\ns@Acrfull}
\glsmfuaddmap{\acrfull}{\Acrfull}
\newcommand*\ns@Acrfull[2][]{%
  \new@ifnextchar[{\@Acrfull{#1}{#2}}%
    {\@Acrfull{#1}{#2}[]}%
}
\def\@Acrfull#1#2[#3]{%
  \Acrfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfull}{\@gls@hyp@opt\ns@ACRfull}
\newcommand*\ns@ACRfull[2][]{%
  \new@ifnextchar[{\@ACRfull{#1}{#2}}%
    {\@ACRfull{#1}{#2}[]}%
}

```

```

\def\@ACRfull#1#2[#3]{%
  \ACRfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*\acrfullpl{\@gls@hyp@opt\ns@acrfullpl}
\newcommand*\ns@acrfullpl[2][]{%
  \new@ifnextchar[{\@acrfullpl{#1}{#2}}%
    {\@acrfullpl{#1}{#2}[ ]}%
}
\def\@acrfullpl#1#2[#3]{%
  \acrfullplfmt{#1}{#2}{#3}%
}
\newrobustcmd*\Acrfullpl{\@gls@hyp@opt\ns@Acrfullpl}
\glsmfuaddmap{\acrfullpl}{\Acrfullpl}
\newcommand*\ns@Acrfullpl[2][]{%
  \new@ifnextchar[{\@Acrfullpl{#1}{#2}}%
    {\@Acrfullpl{#1}{#2}[ ]}%
}
\def\@Acrfullpl#1#2[#3]{%
  \Acrfullplfmt{#1}{#2}{#3}%
}
\newrobustcmd*\ACRfullpl{\@gls@hyp@opt\ns@ACRfullpl}
\newcommand*\ns@ACRfullpl[2][]{%
  \new@ifnextchar[{\@ACRfullpl{#1}{#2}}%
    {\@ACRfullpl{#1}{#2}[ ]}%
}
\def\@ACRfullpl#1#2[#3]{%
  \ACRfullplfmt{#1}{#2}{#3}%
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand*\{acnameformat}[2]{\acronymfont{#1}}
\newtoks\glskeylisttok
\newtoks\glslabeltok
\newtoks\glsshorttok
\newtoks\glslongtok
\newcommand*\{newacronymhook}{-}
\newcommand*\{SetGenericNewAcronym}{-%
  \let\@Gls@entryname\@Gls@acentryname
  \renewcommand{\newacronym}[4][]{%
    \ifdefempty{\@glsacronymlists}%
      {%
        \def\@glo@type{\acronymtype}%
        \setkeys{glossentry}{##1}%
        \DeclareAcronymList{\@glo@type}%
      }%
    }%
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \glslongtok{##4}%
}

```



```

\newacronymhook
\protected@edef\do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\expandonce{\acronymentry{##2}}},%
    sort={\acronymsort{\the\glsshorttok}{\the\glslongtok}},%
    text={\the\glsshorttok},%
    short={\the\glsshorttok},%
    shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
    long={\the\glslongtok},%
    longplural={\the\glslongtok\noexpand\glspluralsuffix},%
    \GenericAcronymFields,%
    \the\glskeylisttok
  }%
}%
\do@newglossaryentry
}%
\renewcommand*\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}}%
\renewcommand*\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
\renewcommand*\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsuppercase{\genacrfullformat{##2}{##3}}}%
\renewcommand*\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsuppercase{\genplacrfullformat{##2}{##3}}}%
\renewcommand*\glsentryfull}[1]{\genacrfullformat{##1}{}}%
\renewcommand*\Glsentryfull}[1]{\Genacrfullformat{##1}{}}%
\renewcommand*\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}%
\renewcommand*\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}%
}
\newcommand*\GenericAcronymFields{\description={\the\glslongtok}}
\newcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{#1}}}
\newcommand*\acronymsort}[2]{#1}
\newcommand*\setacronymstyle}[1]{%
  \ifcsundef{@glsacr@dispstyle@#1}
  {%
    \PackageError{glossaries}{Undefined acronym style ‘#1’}{%
  }%
  }%
  \ifdefempty{@glsacronymlists}%
  {%
    \DeclareAcronymList{\acronymtype}%
  }%
}

```

```

    }%
    \SetGenericNewAcronym
    \GlsUseAcrStyleDefs{#1}%
    \for\@gls@type:=\@glsacronymlists\do{%
        \defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{#1}}%
    }%
}%
}
\newcommand*\newacronymstyle}[3]{%
    \ifcsdef{@glsacr@dispstyle@#1}%
    {%
        \PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
    }%
    {%
        \csdef{@glsacr@dispstyle@#1}{#2}%
        \csdef{@glsacr@styledefs@#1}{#3}%
    }%
}
\newcommand*\renewacronymstyle}[3]{%
    \ifcsdef{@glsacr@dispstyle@#1}%
    {%
        \csdef{@glsacr@dispstyle@#1}{#2}%
        \csdef{@glsacr@styledefs@#1}{#3}%
    }%
    {%
        \PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
    }%
}
\newcommand*\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
{%
    \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
    \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
    \renewcommand*\genacrfullformat}[2]{%
        \glsentrylong{##1}##2\space
        (\protect\firstacronymfont{\glsentryshort{##1}})%
    }%
    \renewcommand*\Genacrfullformat}[2]{%
        \Glsentrylong{##1}##2\space
        (\protect\firstacronymfont{\glsentryshort{##1}})%
    }%
    \renewcommand*\genplacrfullformat}[2]{%
        \glsentrylongpl{##1}##2\space
        (\protect\firstacronymfont{\glsentryshortpl{##1}})%
    }%
    \renewcommand*\Genplacrfullformat}[2]{%
        \Glsentrylongpl{##1}##2\space

```

```

(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
\ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newcommand*{\glsacspace}[1]{%
\settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{##1}})}%
\ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
\ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshort{##1}}##2\space
\glsentrylong{##1}%
}%
}

```

```

\renewcommand*\Genacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshort{##1}}##2\space
  (\glsentrylong{##1})%
}%
\renewcommand*\genplacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
  (\glsentrylongpl{##1})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
  (\glsentrylongpl{##1})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{%
  \GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
}
\newacronymstyle{long-sm-short}%
{%
  \GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*\acrpluralsuffix{\glsacrpluralsuffix}%
}
\newacronymstyle{sc-short-long}%
{%
  \GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
}
\newacronymstyle{sm-short-long}%
{%
  \GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%

```

```

\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sp-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sp-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sp-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\glsacspace{##1}(\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sc-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sc-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sm-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sm-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{short-long-desc}%
{%

```

```

\GlsUseAcrEntryDispStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{sc-short-long-desc}%
{%
\GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
\GlsUseAcrStyleDefs{sc-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{sm-short-long-desc}%
{%
\GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
\GlsUseAcrStyleDefs{sm-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{dua}%
{%
\ifdefempty\glscustomtext
{%
\ifglshaslong{\glslabel}%
{%
\glsifplural
{%
\glscapscase
{%
\glsentrylongpl{\glslabel}\glsinsert
}%
{%
\Glsentrylongpl{\glslabel}\glsinsert
}%
{%
\glsupercase
{\glsentrylongpl{\glslabel}\glsinsert}%
}%
}
}
}
}
}
}
}

```

```

}%
{%
  \glscapscase
  {%
    \glsentrylong{\glslabel}\glsinsert
  }%
  {%
    \Glsentrylong{\glslabel}\glsinsert
  }%
  {%
    \glsupercase
    {\glsentrylong{\glslabel}\glsinsert}%
  }%
}%
}%
{%
  \glsgenentryfmt
}%
}%
{\glscustomtext\glsinsert}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\acrfullfmt}[3]{%
    \glslink[##1]{##2}{\glsentrylong{##2}##3\space
      (\acronymfont{\glsentryshort{##2}})}%
  }%
  \renewcommand*{\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{\Glsentrylong{##2}##3\space
      (\acronymfont{\glsentryshort{##2}})}%
  }%
  \renewcommand*{\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
      \glsupercase{\glsentrylong{##2}##3\space
        (\acronymfont{\glsentryshort{##2}})}%
    }%
  }%
  \renewcommand*{\acrfullplfmt}[3]{%
    \glslink[##1]{##2}{\glsentrylongpl{##2}##3\space
      (\acronymfont{\glsentryshortpl{##2}})}%
  }%
  \renewcommand*{\Acrfullplfmt}[3]{%
    \glslink[##1]{##2}{\Glsentrylongpl{##2}##3\space
      (\acronymfont{\glsentryshortpl{##2}})}%
  }%
  \renewcommand*{\ACRfullplfmt}[3]{%
    \glslink[##1]{##2}{%
      \glsupercase{\glsentrylongpl{##2}##3\space
        (\acronymfont{\glsentryshortpl{##2}})}%
    }%
  }%
  \renewcommand*{\glsentryfull}[1]{%
    \glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*{\Glsentryfull}[1]{%
    \Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*{\glsentryfullpl}[1]{%

```

```

    \glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\Glsentryfullpl[1]{%
    \Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\acronymentry[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort[2]{##1}%
\renewcommand*\acronymfont[1]{##1}%
\renewcommand*\acrpluralsuffix{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
    \GlsUseAcrEntryDispStyle{dua}%
}%
{%
    \GlsUseAcrStyleDefs{dua}%
    \renewcommand*\GenericAcronymFields{}%
    \renewcommand*\acronymentry[1]{\glsentrylong{##1}}%
    \renewcommand*\acronymsort[2]{##2}%
}%
\newacronymstyle{footnote}%
{%
    \ifglsashaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
    \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
    \glshyperfirstfalse
    \renewcommand*\genacrfullformat[2]{%
        \protect\firstacronymfont{\glsentryshort{##1}}##2%
        \protect\footnote{\glsentrylong{##1}}%
    }%
    \renewcommand*\Genacrfullformat[2]{%
        \firstacronymfont{\Glsentryshort{##1}}##2%
        \protect\footnote{\glsentrylong{##1}}%
    }%
    \renewcommand*\genplacrfullformat[2]{%
        \protect\firstacronymfont{\glsentryshortpl{##1}}##2%
        \protect\footnote{\glsentrylongpl{##1}}%
    }%
    \renewcommand*\Genplacrfullformat[2]{%
        \protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
        \protect\footnote{\glsentrylongpl{##1}}%
    }%
    \renewcommand*\acronymentry[1]{\acronymfont{\glsentryshort{##1}}}%
    \renewcommand*\acronymsort[2]{##1}%
    \renewcommand*\acronymfont[1]{##1}%
    \renewcommand*\acrpluralsuffix{\glsacrpluralsuffix}%
    \renewcommand*\acrfullfmt[3]{%
        \glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
            (\glsentrylong{##2})}%
    }%

```



```

\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
  (\Glsentrylong{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
  \glsupercase{\acronymfont{\Glsentryshort{##2}}##3\space
  (\Glsentrylong{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
  (\Glsentrylongpl{##2})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
  (\Glsentrylongpl{##2})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
  \glsupercase{\acronymfont{\Glsentryshortpl{##2}}##3\space
  (\Glsentrylongpl{##2})}}}%
\renewcommand*{\glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\Glsentrylong{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\Glsentrylong{##1})}%
\renewcommand*{\glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\Glsentrylongpl{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\Glsentrylongpl{##1})}%
}
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDisplayStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\Glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}%
\newacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDisplayStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\Glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}%
\newacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{footnote}%
}%

```

```

{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
  }
  \newacronymstyle{footnote-sc-desc}%
  {%
    \GlsUseAcrEntryDispStyle{footnote-sc}%
  }%
  {%
    \GlsUseAcrStyleDefs{footnote-sc}%
    \renewcommand*\GenericAcronymFields{}%
    \renewcommand*\acronymsort}[2]{##2}%
    \renewcommand*\acronymentry}[1]{%
      \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
    }
  \newacronymstyle{footnote-sm-desc}%
  {%
    \GlsUseAcrEntryDispStyle{footnote-sm}%
  }%
  {%
    \GlsUseAcrStyleDefs{footnote-sm}%
    \renewcommand*\GenericAcronymFields{}%
    \renewcommand*\acronymsort}[2]{##2}%
    \renewcommand*\acronymentry}[1]{%
      \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
    }
  \newcommand*\DefineAcronymSynonyms}{%
    \let\acs\acrshort
    \let\Acs\Acrshort
    \let\acsp\acrshortpl
    \let\Acsp\Acrshortpl
    \let\acl\acrlong
    \let\aclp\acrlongpl
    \let\Acl\Acrlong
    \let\Aclp\Acrlongpl
    \let\acf\acrfull
    \let\acfp\acrfullpl
    \let\Acf\Acrfull
    \let\Acfp\Acrfullpl
    \let\ac\gls
    \let\Ac\Gls
    \let\acp\glspl
    \let\Acp\Glspl
  }
  \ifglsacrshortcuts
    \DefineAcronymSynonyms
  \fi

```

```

\newcommand*\SetDefaultAcronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsentryfmt}%
}
\newcommand*\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
\newcommand*\acrlinkfootnote}[3]{%
  \footnote{\glslink[#1]{#2}{#3}}%
}
\newcommand*\acrnolinkfootnote}[3]{%
  \footnote{#3}%
}
\newrobustcmd*\glsdoparenifnotempty}[2]{%
  \protected@edef\gls@tmp{#1}%
  \ifdefempty\gls@tmp
  {}%
  {%
    \ifx\gls@tmp\@gls@default@value
    \else
    \space (#2{#1})%
    \fi
  }%
}
\newcommand*\@gls@deprecated@acr@warning}[3]{%
  \GlossariesWarning{Acronym style option #1 deprecated.
  Use \string\setacronymstyle{#2} with just glossaries.sty or
  use \string\setabbreviationstyle[acronym]{#3} with
  glossaries-extra.sty}%
}
\RequirePackage{glossary-hypernav}
\@gls@loadlist
\@gls@loadlong
\@gls@loadsuper
\@gls@loadtree
\ifx\@glossary@default@style\relax
\else
  \setglossarystyle{\@glossary@default@style}
\fi
\newcommand*\showgloparent}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
}
\newcommand*\showglolevel}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
}
\newcommand*\showglotext}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
}
\newcommand*\showgloplural}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@plural\endcsname
}
\newcommand*\showglofirst}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@first\endcsname
}

```

```

}
\newcommand*\showglofirstpl}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@firstpl\endcsname
}
\newcommand*\showgloftype}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@type\endcsname
}
\newcommand*\showglocounter}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@counter\endcsname
}
\newcommand*\showglouserii}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@userii\endcsname
}
\newcommand*\showglouseriii}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@useriii\endcsname
}
\newcommand*\showglouseriv}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@useriv\endcsname
}
\newcommand*\showglouserv}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@userv\endcsname
}
\newcommand*\showglouservi}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@uservi\endcsname
}
\newcommand*\showgloname}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@name\endcsname
}
\newcommand*\showglodesc}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@desc\endcsname
}
\newcommand*\showglodescplural}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@descplural\endcsname
}
\newcommand*\showglosort}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@sort\endcsname
}
\newcommand*\showglosymbol}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@symbol\endcsname
}
\newcommand*\showglosymbolplural}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@symbolplural\endcsname
}
\newcommand*\showgloshort}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@short\endcsname
}
\newcommand*\showglolong}[1]{%

```

```

    \expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname
  }
  \newcommand*{\showgloindex}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname
  }
  \newcommand*{\showgloflag}[1]{%
    \expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname
  }
  \newcommand*{\showglooclist}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@oclist\endcsname
  }
  \newcommand*{\showglofield}[2]{%
    \csshow{glo@\glsdetoklabel{#1}@#2}%
  }
  \newcommand*{\showacronymlists}{%
    \show@\glsacronymlists
  }
  \newcommand*{\showglossaries}{%
    \show@\glo@types
  }
  \newcommand*{\showglossaryin}[1]{%
    \expandafter\show\csname @glo@type@#1@in\endcsname
  }
  \newcommand*{\showglossaryout}[1]{%
    \expandafter\show\csname @glo@type@#1@out\endcsname
  }
  \newcommand*{\showglossarytitle}[1]{%
    \expandafter\show\csname @glo@type@#1@title\endcsname
  }
  \newcommand*{\showglossarycounter}[1]{%
    \expandafter\show\csname @glo@type@#1@counter\endcsname
  }
  \newcommand*{\showglossaryentries}[1]{%
    \expandafter\show\csname glo@list@#1\endcsname
  }
}

```

Change History

1.01 – 2007-05-17	
General: Added range facility in format key	129
\writeist: Added spaces after \delimN and \delimR in ist file	184
1.04 – 2007-08-03	
General: Added \glstextformat	117
1.05 – 2007-08-10	
\glossarysection: added	
\@mkboth to	
\glossarysection	54
\gls@defglossaryentry:	
Changed the default value of the sort key to just the value of the name key	99
1.07 – 2007-09-13	
\@gls@link: fixed bug caused by \theglsentrycounter setting the page number too soon	127
\glsadd: fixed bug caused by \theglsentrycounter setting the page number too soon	181
1.08 – 2007-10-13	
General: Added babel support	48
altlistgroup: changed altlistgroup style to use	
\glsgetgrouptitle	289
listgroup: changed listgroup style to use	
\glsgetgrouptitle	288
1.1 – 2008-02-22	
General: amsgen now loaded (\new@ifnextchar needed)	6
\@glossarysection: numbered sections and auto label added	55
\@gls@tmpb: changed \toksdef to \newtoks	131
\@gls@toc: numberline added	57
\@p@glossarysection: numbered sections and auto label added	56
numberedsection:	
numberedsection package option added	10
numberline: numberline option added	10
\setglossarysection: new	55
translate: translate option added	30
1.12 – 2008-03-08	
General: added check for \hypertarget separate to \hyperlink (memoir defines \hyperlink but not \hypertarget)	139
\@GLSpl: now uses	
\glsentrydescplural and \glsentrysymbolplural instead of \glsentrydesc and \glsentrysymbol	145
\@Glspl0: now uses	
\glsentrydescplural and \glsentrysymbolplural instead of \glsentrydesc and \glsentrysymbol	144
\@glspl0: now uses	
\glsentrydescplural and \glsentrysymbolplural instead of \glsentrydesc and \glsentrysymbol	143
descriptionplural: new	79
\gls@defglossaryentry:	
Changed default first plural to be first key with s appended (was text key with s appended)	98
descriptionplural support added	98
symbolplural support added	98
\Glsentrydescplural: New	173
\glsentrydescplural: New	173
\Glsentrysymbolplural: New	174
\glsentrysymbolplural: New	174
symbolplural: new	80
1.13 – 2008-05-10	
General: fixed bug that ignored 3rd parameter	146–155
\ACRfullpl: new	255
\Acrfullpl: new	254
\acrfullpl: new	254
\acrpluralsuffix: New	251
\gls@defglossaryentry:	
Changed default first value	98

Changed default firstplural value	98	\@GLspl@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	144
Removed restriction on only using \newglossaryentry in the preamble	103	\@gls@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	141
\newacronym: Removed restriction on only using \newacronym in the preamble	251	\@glsdisp: Test glossary type is \acronymtype in addition to checking if footnote option has been used	145
1.14 – 2008-06-17		\@glspl@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	143
General: added nonumberlist key to \printglossary	238	\@gls@defglossaryentry: Changed def to let	98
added numberedsection key to \printglossary	237	1.17 – 2008-12-26	
\@gls@hypergroup: new	283	General: added xindy support	33
\firstacronymfont: new	255	\@do@esc@wrglossary: new	209
\glsautoprefix: new	10	\@do@seeglossary: new	214
\glsnavhyperlink: changed \edef to \protected@edef	280	\@glo@storeentry: new	105
\glsnavhypertarget: added write to aux file	282	\@gls@glossary: changed definition to use \index instead of \@index	205
\glsnavigation: changed to only use labels for groups that are present	283	\@glsdefaultplural: new	84
1.15 – 2008-08-15		\@glsdefaultsort: new	84
\@gls@link: added \glslabel	127	\@gls@hypernumber: new	247
\@gls@defglossaryentry: check for \@glo@first in description	102	\@glsnoname: new	83
check for \@glo@text in symbol	103	\@glsnonextpages: new	238
\@gls@hypergroup: new	283	\@gls@defglossaryentry: added nonumberlist key	99
\@glsnavhypertarget: added check if rerun required	282	added parent key	99
\@glssettoctitle: new	47	added see key	99
\printglossary: changed the way the TOC title is set	216	Stored main part of entry format when entry is defined	103
1.16 – 2008-08-27		\@gls@suffixF: new	52
\@GLS@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	142	\@gls@suffixFF: new	52
\@GLSpl: Test glossary type is \acronymtype in addition to checking if footnote option has been used	145	\@gls@wrglossary: modified to allow for xindy support	206
\@Gls@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	141	\@gls@hyperlink: new	180
		\@gls@hypernumber: modified to allow material to be attached to location	247

<code>\glsnavhyperlink</code> : replaced	1.46 – 2021-09-20
<code>\hyperlink</code> to <code>\@glslink</code>	280
<code>\glsnavhypertarget</code> : replaced	General: changed <code>\edef</code> to
<code>\hypertarget</code> with	<code>\protected@edef</code>
<code>\@glsstarget</code>	282
<code>\glssee</code> : new	214
<code>\glsseeformat</code> : new	215
<code>\glsSetSuffixF</code> : new	52
<code>\glsSetSuffixFF</code> : new	53
<code>\ifglsxindy</code> : new	33
<code>\istfilename</code> : added xindy support	51
<code>\newglossarystyle</code> : made	<code>\@printglossary</code> : changed <code>\xdef</code>
<code>\newglossarystyle long</code>	to <code>\protected@xdef</code>
<code>nonumberlist</code> : new	81
<code>\nopostdesc</code> : new	50
<code>order</code> : order package option	<code>\gls@defglossaryentry</code> : changed
added	<code>\xdef</code> to
<code>parent</code> : new	<code>\protected@xdef</code>
<code>\printglossary</code> : added check to	<code>\glsaddprotectedpagefmt</code> :
determine if <code>\printglossary</code>	changed <code>\eappto</code> to
is already defined	<code>\protected@eappto</code>
added print language to aux	208
file	2.01 – 2009 May 30
<code>see</code> : new	80
<code>\writeist</code> : added xindy support	<code>\@gls@link</code> : moved
184	<code>\@do@wrglossary</code> before term
1.18 – 2009-01-14	is displayed to prevent
<code>\@gls@loadlist</code> : new	unwanted whatsit
13	127
<code>\@gls@loadlong</code> : new	<code>\forallglossaries</code> : replaced
12	<code>\ifthenelse</code> with <code>\ifx</code>
<code>\@gls@loadsuper</code> : new	66
12	<code>\forglentries</code> : replaced
<code>\@gls@loadtree</code> : new	<code>\ifthenelse</code> with <code>\ifx</code>
13	67
<code>\gls@defglossaryentry</code> :	<code>\glsdefmain</code> : new
Changed default value of sort	19
to <code>\@glsdefaultsort</code>	<code>\glsdescwidth</code> : changed
99	<code>\linewidth</code> to <code>\hsize</code>
moved sort sanitization to	291, 314
<code>\newglossaryentry</code>	<code>\glslistdottedwidth</code> : changed
103	<code>\linewidth</code> to <code>\hsize</code>
<code>\glsstarget</code> : new	290
241	<code>\glspagelistwidth</code> : changed
<code>nolist</code> : new	<code>\linewidth</code> to <code>\hsize</code>
13	291, 314
<code>nolong</code> : new	<code>nomain</code> : added <code>nomain</code> package
12	option
<code>nostyles</code> : new	20
13	<code>\writeist</code> : removed <code>item_02</code> -
<code>nosuper</code> : new	no such <code>makeindex</code> key
13	188
<code>notree</code> : new	2.02 – 2007-07-13
13	<code>\@printglossary</code> : suppressed
<code>\oldacronym</code> : new	warning globally rather than
250	locally
<code>sort</code> : moved sanitization to	219
<code>\newglossaryentry</code>	2.02 – 2009-07-13
79	<code>\glossarysection</code> : changed
1.19 – 2009-03-02	<code>\@mkboth</code> to <code>\glossarymark</code>
<code>\glsclearpage</code> : new	54
57	<code>\gls@defglossaryentry</code> : New
<code>\glsdisp</code> : new	54
145	2.03 – 2009-09-23
	<code>\@GLS@</code> : Added check for
	<code>hyperfirst</code>
	142
	<code>\@GLSp1</code> : Added check for
	<code>hyperfirst</code>
	145
	<code>\@GLs@</code> : Added check for
	<code>hyperfirst</code>
	141
	<code>\@GLsp1@</code> : Added check for
	<code>hyperfirst</code>
	144
	<code>\@gls@</code> : Added check for
	<code>hyperfirst</code>
	141

<code>\@gls@link</code> : new	125	<code>\gls@defglossaryentry</code> : added	
<code>\@gls@link</code> : added <code>\leavevmode</code>	127	user1-6 keys	99
Moved entry existence check to		<code>\glsadd</code> : fixed bug that ignored	
avoid duplicate code	127	counter	181
<code>\@glsdisp</code> : Added check for		<code>\Glsentryuseri</code> : new	176
<code>hyperfirst</code>	145	<code>\glsentryuseri</code> : new	176
<code>\@glspl@</code> : Added check for		<code>\Glsentryuserii</code> : new	176
<code>hyperfirst</code>	143	<code>\glsentryuserii</code> : new	176
<code>\gls glossarymark</code> : Added check		<code>\Glsentryuseriii</code> : new	176
to see if it's already defined	54	<code>\glsentryuseriii</code> : new	176
<code>hyperfirst</code> : new	31	<code>\Glsentryuseriv</code> : new	177
2.04 – 2009-11-10		<code>\glsentryuseriv</code> : new	176
<code>\@GLS@</code> : Changed test to check if		<code>\Glsentryuserv</code> : new	177
glossary type has been		<code>\glsentryuserv</code> : new	177
identified as a list of		<code>\Glsentryuservi</code> : new	177
acronyms	142	<code>\glsentryuservi</code> : new	177
<code>\@GLSpl@</code> : Changed test to check if		<code>\ns@newglossary</code> : added check to	
glossary type has been		determine if	
identified as a list of		<code>\gls@<i>type</i>@display</code> and	
acronyms	145	<code>\gls@<i>type</i>@displayfirst</code>	
<code>\@Gls@</code> : Changed test to check if		have been defined.	76
glossary type has been		<code>\SetAcronymLists</code> : new	22
identified as a list of		<code>\SetDefaultAcronymDisplayStyle</code> :	
acronyms	141	new	270
<code>\@Glspl@</code> : Changed test to check		<code>\SetDefaultAcronymStyle</code> : new	270
if glossary type has been		<code>\SetDescriptionAcronymDisplayStyle</code> :	
identified as a list of		new	271
acronyms	144	<code>\SetDescriptionDUAAcronymDisplayStyle</code> :	
<code>\@glossaryentryfield</code> : new	104	new	271
<code>\@glossarysubentryfield</code> : new	104	<code>\SetDescriptionFootnoteAcronymDisplayStyle</code> :	
<code>\@gls@</code> : Changed test to check if		new	271
glossary type has been		<code>\SetDUADisplayStyle</code> : new	272
identified as a list of		<code>\SetFootnoteAcronymDisplayStyle</code> :	
acronyms	141	new	271
<code>\@glsacronymlists</code> : new	21	<code>\SetSmallAcronymDisplayStyle</code> :	
<code>\@glsdisp</code> : Changed test to check		new	272
if glossary type has been		2.05 – 2010-02-06	
identified as a list of		<code>\@glsdisp</code> : Added closing brace.	
acronyms	145	Patch provided by Sergiu	
<code>\@glspl@</code> : Changed test to check		Dotenco	145
if glossary type has been		Removed spurious brace. Patch	
identified as a list of		provided by Sergiu Dotenco	145
acronyms	143	<code>\writeist</code> : Added <code>\string</code>	
<code>\@newglossaryentryposthook</code> :		before opening and closing	
new	104	braces. Patch provided by	
<code>\@newglossaryentryprehook</code> :		Segiu Dotenco	189
new	104	2.06 – 2010-06-14	
<code>acronymlists</code> : new	22	<code>\altnewglossary</code> : new	76
<code>\DeclareAcronymList</code> : new	21	<code>\CustomAcronymFields</code> : new	273
<code>\DefineAcronymSynonyms</code> : new	268	<code>\CustomNewAcronymDef</code> : new	273

\SetCustomDisplayStyle: new .	273	added \glsnextpages	219
\SetCustomStyle: new	273	make toctitle default to title .	218
2.07 – 2010-07-10		\@xdyattributelist: new	58
General: glsadd format key stored		\acrfootnote: new	270
in \@glsnumberformat (was		\ACRfull: added starred version	254
mistakenly stored in		\Acrfull: added starred version	253
\@glo@format)	180	\acrfull: added starred version	253
3.0 – 2010-07-12		\ACRfullpl: added starred	
\@makeglossary: Added check for		version	255
savewrites	194	\Acrfullpl: added starred	
\@gls@wrglossary: modified to		version	254
take into account savewrites	206	\acrfullpl: added starred	
3.0 – 2010/03/31		version	254
\@set@glo@numformat: added 4th		\acrlinkfootnote: new	270
argument	129	\acrnofootnote: new	271
3.0 – 2011-04-02		compatible-2.07:	
General: added prefix to		compatible-2.07 option added	38
hyperlink	248	counter: replaced \@ifundefined	
etoolbox now loaded	6	with \ifcsundef	80
replaced \@ifundefined with		counterwithin: new	14
\ifcsundef	46, 49, 123, 236	entrycounter: new	14
\@do@esc@wrglossary: added		glossaryentry: new	14
check for hyper location		\glossarysection: replaced	
prefix	212	\@ifundefined with	
modified to use new format .	209	\ifcsundef	54
\@glossarysec: replaced		\glossarystyle: replaced	
\@ifundefined with		\@ifundefined with	
\ifcsundef	10	\ifcsundef	246
\@do@seeglossary: Sanitize and		glossarysubentry: new	15
escape cross-referencing		\gls@codepage: replaced	
information	214	\@ifundefined with	
\@gls@counterwithin: new	15	\ifcsundef	34
\@gls@ifinlist: new	58	\gls@defglossaryentry: added	
\@gls@link: added		\@gls@defsot	103
\@gls@saveentrycounter .	127	added short and long keys . .	99
added \@gls@setsort	127	replaced \@ifundefined with	
\@gls@saveentrycounter: new .	128	\ifcsundef	99
\@gls@setupsort@def: new	17	\gls@doclearpage: replaced	
\@gls@setupsort@standard: new	16	\@ifundefined with	
\@gls@setupsort@use: new	17	\ifcsundef	56
\@gls@xdy@locationlist: new .	61	\glsadd: added	
\@glslink: replaced		\@gls@saveentrycounter .	181
\@ifundefined with		\GlsAddXdyCounters: new	58
\ifcsundef	139	\glsentrycounterlabel: new .	240
\@glsnextpages: new	238	\glsentryitem: new	240
\@print@glossary: replaced		\Glsentrylong: new	178
\@ifundefined with		\glsentrylong: new	178
\ifcsundef	220	\Glsentrylongpl: new	178
\@printglossary: added		\glsentrylongpl: new	178
\currentglossary	219	\Glsentryshort: new	177

<code>\glstryshort</code> : new	177	replaced <code>\@ifundefined</code> with	
<code>\Glsentryshortpl</code> : new	178	<code>\ifcsundef</code>	76
<code>\glstryshortpl</code> : new	177	<code>\oldacronym</code> : replaced	
<code>\glsgrouptitle</code> : replaced		<code>\@ifundefined</code> with	
<code>\@ifundefined</code> with		<code>\ifcsundef</code>	250
<code>\ifcsundef</code>	244	<code>\printglossary</code> : replaced	
<code>\glsglossarymark</code> : replaced		<code>\@ifundefined</code> with	
<code>\@ifundefined</code> with		<code>\ifcsundef</code>	216
<code>\ifcsundef</code>	54	<code>savewrites</code> : new	38
<code>\glshyperlink</code> : changed default		<code>see</code> : added	
from <code>\glstryname</code> to		<code>\@glo@seeautonumberlist</code>	80
<code>\glstrytext</code>	180	<code>seeautonumberlist</code> : new	12
<code>\glshyphnumber</code> : replaced		<code>\SetDescriptionFootnoteAcronymDisplayStyle</code> :	
<code>\@ifundefined</code> with		expanded options link	
<code>\ifcsundef</code>	247	options	271
<code>\glstnumberformat</code> : replaced		<code>\setentrycounter</code> : added	
<code>\@ifundefined</code> with		optional argument	245
<code>\ifcsundef</code>	53	<code>short</code> : new	83
<code>\glstrefentry</code> : new	240	<code>shortplural</code> : new	83
<code>\glstresetsubentrycounter</code> :		<code>\showacronymlists</code> : new	278
new	239	<code>\showglocounter</code> : new	275
<code>\glstseeitem</code> : hyperlink uses		<code>\showgloDESC</code> : new	276
<code>\glstseeitemformat</code> instead of		<code>\showgloDESCplural</code> : new	277
<code>\glstryname</code>	215	<code>\showglofirst</code> : new	274
<code>\glstseeitemformat</code> : new	215	<code>\showglofirstpl</code> : new	275
<code>\glstsortnumberfmt</code> : new	16	<code>\showgloflag</code> : new	278
<code>\glststepentry</code> : new	239	<code>\showgloindex</code> : new	278
<code>\glststepsubentry</code> : new	239	<code>\showglolevel</code> : new	274
<code>\glstsubentrycounterlabel</code> :		<code>\showglongame</code> : new	276
new	240	<code>\showgloparent</code> : new	274
<code>\glstsubentryitem</code> : new	240	<code>\showgloplural</code> : new	274
<code>\ifglossaryexists</code> : replaced		<code>\showglosort</code> : new	277
<code>\@ifundefined</code> with		<code>\showglossaries</code> : new	278
<code>\ifcsundef</code>	67	<code>\showglossarycounter</code> : new	279
<code>\ifglstryexists</code> : replaced		<code>\showglossaryentries</code> : new	279
<code>\@ifundefined</code> with		<code>\showglossaryin</code> : new	279
<code>\ifcsundef</code>	68	<code>\showglossaryout</code> : new	279
<code>\istfile</code> : deprecated	204	<code>\showglossarytitle</code> : new	279
<code>long</code> : new	83	<code>\showglosymbol</code> : new	277
<code>longplural</code> : new	83	<code>\showglosymbolplural</code> : new	277
<code>\newglossaryentry</code> : replaced		<code>\showglotext</code> : new	274
<code>\DeclareRobustCommand</code> with		<code>\showglotype</code> : new	275
<code>\newrobustcmd</code>	86	<code>\showglouserI</code> : new	275
<code>\newglossarystyle</code> : replaced		<code>\showglouserII</code> : new	275
<code>\@ifundefined</code> with		<code>\showglouserIII</code> : new	276
<code>\ifcsundef</code>	246	<code>\showglouserIV</code> : new	276
<code>nonumberlist</code> : now boolean	81	<code>\showglouserV</code> : new	276
<code>\ns@newglossary</code> : added		<code>\showglouserVI</code> : new	276
<code>\@gls@defsortcount</code>	76	<code>sort</code> : new	15
		<code>subentrycounter</code> : new	15

theglossary: replaced		\glsname: made robust	150
\@ifundefined with		\GLSpl: made robust	144
\ifcsundef	241	\Glspl: made robust	143
\writeist: added xindy-only		\glspl: made robust	142
macro definitions to glossary		\GLSplplural: made robust	149
open tag	186	\GLSsymbol: made robust	154
modified to support new		\Glsymbol: made robust	153
format	184	\glsymbol: made robust	153
3.01 – 2011-04-12		\GLSsymbolplural: made robust	155
General: made robust	142	\Glsymbolplural: made robust	154
\@glswritefiles: added check		\glsymbolplural: made robust	154
for empty glossaries	204	\GLstext: made robust	147
\ACRfull: made robust	254	\glstext: made robust	146
\Acrfull: made robust	253	\GLSuseri: made robust	156
\acrfull: made robust	253	\Glsuseri: made robust	155
\acrfullformat: removed		\glsuseri: made robust	155
\acronymfont as it should		\GLSuserii: made robust	157
already be set in the second		\Glsuserii: made robust	156
argument.	252	\glsuserii: made robust	156
\ACRfullpl: made robust	255	\GLSuseriii: made robust	158
\Acrfullpl: made robust	254	\Glsuseriii: made robust	157
\acrfullpl: made robust	254	\glsuseriii: made robust	157
\ACRlong: made robust	166	\GLSuseriv: made robust	159
\Acrlong: made robust	166	\Glsuseriv: made robust	158
\acrlong: made robust	165	\glsuseriv: made robust	158
\ACRlongpl: made robust	168	\GLSuseriv: made robust	160
\Acrlongpl: made robust	168	\Glsuseriv: made robust	159
\acrlongpl: made robust	167	\glsuseriv: made robust	159
\ACRshort: made robust	162	\GLSuserivi: made robust	161
\Acrshort: made robust	162	\Glsuserivi: made robust	160
\acrshort: made robust	161	\glsuserivi: made robust	160
\ACRshortpl: made robust	164	3.02 – 2012-05-19	
\Acrshortpl: made robust	164	\glsnumlistlastsep: new	180
\acrshortpl: made robust	163	\glsnumlistsep: new	180
\Gls: made robust	141	3.02 – 2012-05-21	
\glsadd: made robust	181	\@do@wrglossary: changed	
\glsaddall: made robust	181	\@glslocref to	
\GLSdesc: made robust	152	\theglsentrycounter	212
\Glsdesc: made robust	151	\do@wrglossary: changed	
\glsdesc: made robust	151	\do@wr@glossary to test for	
\GLSdescplural: made robust	153	indexonlyfirst option; put old	
\Glsdescplural: made robust	152	\do@wr@glossary code into	
\glsdescplural: made robust	152	\@do@wrglossary	206
\glsfirst: made robust	147	\@gls@missingnumberlist: new	84
\GLSfirstplural: made robust	150	\@glswritefiles: added check	
\Glsfirstplural: made robust	149	for existence of token in case	
\glsfirstplural: made robust	149	\makeglossaries has been	
\glslink: made robust	125	omitted	204
\GLSname: made robust	151	\@printglossary: add a way to	
\Glsname: made robust	150	fetch current entry label	219

<code>\gls@defglossaryentry</code> : added numberlist element	102	<code>\glspostinline</code> : replaced “.” with <code>\glspostdescription</code>	286
<code>\gls@save@numberlist</code> : new . .	216	index: added check for <code>glsnogroupskip</code>	329
<code>\gls@wrglossary</code> : added check for glossary file defined . . .	206	list: added check for <code>glsnogroupskip</code>	288
<code>\glsdisplaynumberlist</code> : new .	179	long: added check for <code>glsnogroupskip</code>	292
<code>\glsentrycounter</code> : set default value	128	long3col: added check for <code>glsnogroupskip</code>	293
<code>\Glsentryfull</code> : fixed bug (replaced <code>\glsentryshortpl</code> with <code>\glsentryshort</code>)	178	long4col: added check for <code>glsnogroupskip</code>	295
<code>\glsentryfullpl</code> : fixed bug (replaced <code>\glsentryshort</code> with <code>\glsentryshortpl</code>) . .	178	longragged: added check for <code>glsnogroupskip</code>	303
<code>\glsentrynumberlist</code> : new . . .	179	longragged3col: added check for <code>glsnogroupskip</code>	304
<code>\glsmoveentry</code> : new	104	nogroupskip: new	14
<code>\glsresetentrycounter</code> : new .	239	nopostdot: new	13
<code>\ifglshaschildren</code> : new	70	super: added check for <code>glsnogroupskip</code>	314
<code>\ifglshasparent</code> : new	70	super3col: added check for <code>glsnogroupskip</code>	316
indexonlyfirst: new	31	super4col: added check for <code>glsnogroupskip</code>	318
<code>\makeglossaries</code> : added list parser	199	superragged: added check for <code>glsnogroupskip</code>	322
<code>\renewglossarystyle</code> : new . . .	246	superragged3col: added check for <code>glsnogroupskip</code>	324
savenumberlist: new	12	tree: added check for <code>glsnogroupskip</code>	331
<code>\showglossaryentries</code> : fixed misspelt command	279	treenoname: added check for <code>glsnogroupskip</code>	332
<code>\SmallNewAcronymDef</code> : fixed broken short and long plural	272		
ucmark: new	14		
3.03 – 2012/09/21			
General: allow title to set toctitle	236		
<code>\@gls@sanitizesort</code> : new	25	3.04 – 2012-11-11	
<code>\@gls@setupsort@standard</code> : used <code>\@gls@sanitizesort</code>	16	<code>altlist</code> : replaced <code>\newline</code> with paragraph break	289
<code>\@printglossary</code> : allow title to override default toctitle . . .	218	3.04 – 2012-11-18	
<code>altlongragged4col</code> : added check for <code>glsnogroupskip</code>	306	General: Added check for doc package	6
<code>altsuperragged4col</code> : added check for <code>glsnogroupskip</code> . .	325	added datatool-base as a required package	50
<code>almtree</code> : added check for <code>glsnogroupskip</code>	335	added local key	124
<code>\glsinlinedescformat</code> : new . .	286	<code>\@do@wrglossary</code> : changed <code>\theglsentrycounter</code> back to <code>\@glslocref</code>	212
<code>\glsinlineemptydescformat</code> : new	286	<code>\@do@esc@wrglossary</code> : modified to compensate for possible incorrect page number	211
<code>\glsinlinenameformat</code> : new . .	286	<code>\@gls@esc@sdq</code> : unsanitize <code>\gls@numberpage</code> , <code>\gls@alphpage</code> ,	
<code>\glsinlinepostchild</code> : new . . .	286		
<code>\glsinlinesubdescformat</code> : new	286		
<code>\glsinlinesubnameformat</code> : new	286		

\gls@Alphpage and			
\gls@romanpage	130	3.07 – 2013-07-05	
\@print@glossary: Moved aux			
write to end of document to			
prevent unwanted whatsit			
occurring here.	220		
\gls@Alphpage: new	207		
\gls@alphpage: new	207		
\gls@disablepagerefexpansion:			
new	207		
\gls@numberpage: new	207		
\gls@protected@pagefmts: new	207		
\gls@romanpage: new	207		
\glsdefmain: added check for doc			
package	19		
\glsorg@endtheglossary: new	6		
\glsorg@theglossary: new	6		
\PrintChanges: new	6	3.08a – 2013-08-30	
3.05 – 2013-04-21			
\@do@esc@wrglossary: add			
Roman case. Fixed bugs in			
the else statements	211		
\@gls@link: added check for			
“nohypertypes”	127		
\gls@protected@pagefmts:			
added Roman to list	207		
\gls@Romanpage: new	207		
\glsgetgrouplabel: fixed bug			
(typo in \equal)	245		
\glsmcols: new	308		
mcolalttree: replaced ‘2’ with			
\glsmcols	312	3.08a – 2013-09-28	
mcolindex: replaced ‘2’ with			
\glsmcols	308		
mcolindexspannav: replaced ‘2’			
with \glsmcols	309		
mcoltree: replaced ‘2’ with			
\glsmcols	309		
mcoltreenoname: replaced ‘2’			
with \glsmcols	311		
mcoltreesspannav: replaced ‘2’			
with \glsmcols	310		
\nopostdesc: made robust	50		
3.05 – 2013/04/21			
\@gls@nohyperlist: new	22		
\GlsDeclareNoHyperList: new	22		
nohypertypes: new	23		
3.06 – 2013/06/17			
\@xdy@main@language: Changed			
back to using \language	33		
\findrootlanguage: Obsoleted	65		
\@gls@link: fixed bug that failed			
to find entry in list	127		
\glossarypreamble: modified to			
work with			
\setglossarypreamble	54		
\gls@docclearpage: added check			
for openright	56		
\glspostdescription: Added			
spacefactor code	13		
\GlsSetXdyCodePage: Added			
check for fontspec	66		
\SetDescriptionAcronymDisplayStyle:			
now using			
\glsdoparenifnotempty	271		
\setglossarypreamble: new	54		
3.08a – 2013-08-30			
altlist: updated altlist style to			
use \glossentry and			
\subglossentry	289		
inline: updated inline style to			
use \glossentry and			
\subglossentry	284		
list: updated list style to use			
\glossentry and			
\subglossentry	288		
listdotted: updated listdotted			
style to use \glossentry and			
\subglossentry	290		
3.08a – 2013-09-28			
General: added nogroupskip key			
to \printglossary	237		
removed definition of			
\@glossaryentryfield	371		
removed definition of			
\@glossarysubentryfield	371		
\@glo@storeentry: no longer			
need to check for special			
characters in any of the fields			
other than sort	105		
updated for \glossentry	105		
\@glossaryentryfield: switched			
to \glossentry	104		
\@glossarysubentryfield:			
switched to \subglossentry	104		
\@gls@sanitizedesc: deprecated	24		
altlongragged4col: updated to			
use \glossentry and			
\subglossentry	306		

alttree: updated to use		superragged: updated to use	
\glossentry and		\glossentry and	
\subglossentry	334	\subglossentry	321
\compatibleglossentry: new ..	241	tree: updated to use	
\compatiblesubglossentry:		\glossentry and	
new	243	\subglossentry	330
\glossaryentryfield:		3.09a – 2013-10-09	
deprecated	243	\@gls@assign@symbolplural@field:	
\glossarystyle: deprecated ..	246	new	25
\glossentry: new	241	\@gls@default@value: new ...	79
\Glossentrydesc: new	242	\Glsentrydesc: made robust ..	172
\glossentrydesc: new	242	\Glsentrydescplural: made	
\Glossentryname: new	242	robust	173
\glossentryname: new	241	\Glsentryfirst: made robust ..	174
\Glossentrysymbol: new	242	\Glsentryfirstplural: made	
\glossentrysymbol: new	242	robust	175
\gls@assign@desc@field: new ..	24	\Glsentryfull: made robust ..	178
\gls@assign@descplural@field:		\Glsentryfullpl: made robust	179
new	25	\Glsentrylong: made robust ..	178
\gls@assign@field: new	86	\Glsentrylongpl: made robust	178
\gls@ifnotmeasuring: new ...	107	\Glsentryname: made robust ..	171
\glsaddallunused: new	182	\Glsentryplural: made robust	173
\glsexpandfields: new	86	\Glsentryshort: made robust ..	177
\glsnoexpandfields: new	86	\Glsentryshortpl: made robust	178
\glssee: made robust	214	\Glsentrysymbol: made robust	174
\glsseeformat: made robust ..	215	\Glsentrysymbolplural: made	
\glsseeitem: made robust	215	robust	174
\glsseelist: made robust	215	\Glsentrytext: made robust ..	173
\ifglsdescsuppressed: new ..	71	\Glsentryuseri: made robust ..	176
\ifglsdesc: new	71	\Glsentryuserii: made robust	176
\ifglsymbol: new	71	\Glsentryuseriii: made robust	176
index: added paragraph break at		\Glsentryuseriv: made robust	177
end of environment	328	\Glsentryuserv: made robust ..	177
updated to use \glossentry		\Glsentryuservi: made robust	177
and \subglossentry	329	\glstextup: new	251
long: updated to use		\ifglsymbol: changed test	
\glossentry and		to check for	
\subglossentry	291	\@gls@default@symbol	71
longragged: updated to use		3.10a – 2013-09-28	
\glossentry and		\gls@assign@type@field: new ..	24
\subglossentry	303	3.10a – 2013-10-13	
longragged3col: updated to use		\@gls@keymap: new	89
\glossentry and		\@gls@provide@newglossary:	
\subglossentry	304	new	74
\setglossarystyle: new	245	\@gls@writedef: new	89
\setglossentrycompatibility:		\@glsdefaultplural: Obsolete ..	84
new	243	\@glsnodesc: new	83
\subglossentry: new	243	\@print@glossary: Added	
		providecommand code to aux	
		file	220

<code>\gls@defglossaryentry:</code>	
Changed to using	
<code>\@gls@default@value</code> ..	98, 99
new	98
<code>\gls@writedefhook:</code> new	97
<code>\makeglossaries:</code> Added	
providecommand code to aux	
file	198
<code>\new@glossaryentry:</code> new	87
<code>\ns@newglossary:</code> added	
<code>\@gls@provide@newglossary</code>	76
3.11a – 2013-10-15	
General: added <code>\glslabel</code> ,	
<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\glsinsert</code> and	
<code>\glscustomtext</code>	161–169
changed to just use	
<code>\Glsentrydescplural</code>	153
changed to just use	
<code>\glsentrydescplural</code> ..	152, 153
changed to just use	
<code>\Glsentrydesc</code>	152
changed to just use	
<code>\glsentrydesc</code>	151, 152
changed to just use	
<code>\Glsentryfirstplural</code> ...	150
changed to just use	
<code>\glsentryfirstplural</code>	149, 150
changed to just use	
<code>\Glsentryfirst</code>	148
changed to just use	
<code>\glsentryfirst</code>	147, 148
changed to just use	
<code>\Glsentryname</code>	151
changed to just use	
<code>\glsentryname</code>	150, 151
changed to just use	
<code>\Glsentryplural</code>	149
changed to just use	
<code>\glsentryplural</code>	148, 149
changed to just use	
<code>\Glsentrysymbolplural</code> ..	155
changed to just use	
<code>\glsentrysymbolplural</code>	154, 155
changed to just use	
<code>\Glsentrysymbol</code>	154
changed to just use	
<code>\glsentrysymbol</code>	153, 154
Changed to just use	
<code>\Glsentrytext</code>	147
changed to just use	
<code>\glsentrytext</code>	146
changed to just use	
<code>\Glsentryuseriii</code>	158
changed to just use	
<code>\glsentryuseriii</code> ...	157, 158
changed to just use	
<code>\Glsentryuserii</code>	157
changed to just use	
<code>\glsentryuserii</code>	156, 157
changed to just use	
<code>\Glsentryuseriv</code>	159
changed to just use	
<code>\glsentryuseriv</code>	158, 159
changed to just use	
<code>\Glsentryuseri</code>	156
changed to just use	
<code>\glsentryuseri</code>	155, 156
changed to just use	
<code>\Glsentryuservi</code>	161
changed to just use	
<code>\glsentryuservi</code>	160, 161
changed to just use	
<code>\Glsentryuserv</code>	160
changed to just use	
<code>\glsentryuserv</code>	159, 160
Now requires <code>textcase</code>	5
<code>\@ACRlong:</code> added <code>\glslabel</code> ,	
<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\glsinsert</code> and	
<code>\glscustomtext</code>	366
<code>\@ACRshort:</code> added <code>\glslabel</code> ,	
<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\glsinsert</code> and	
<code>\glscustomtext</code>	365
<code>\@Acrlong:</code> added <code>\glslabel</code> ,	
<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\glsinsert</code> and	
<code>\glscustomtext</code>	366
<code>\@Acrshort:</code> added <code>\glslabel</code> ,	
<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\glsinsert</code> and	
<code>\glscustomtext</code>	364
<code>\@GLS@:</code> add <code>\glslabel</code> ,	
<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\glscustomtext</code> and	
<code>\glsinsert</code>	142
change to using <code>\glsentryfmt</code>	
style commands	142

removed \MakeUppercase (now moved to \glsentryfmt) ..	142	\@glspl@: add \glslabel,	
\@GLSpl: add \glslabel,		\glsifplural, \glscapscase,	
\glsifplural, \glscapscase,		\glsinsert	143
\glsinsert	144	change to using \glsentryfmt	
change to using \glsentryfmt		style commands	143
style commands	145	acronymlists: replaced	
removed \MakeUppercase as now dealt with in		\@addtoacronymlists with	
\glsentryfmt	145	\DeclareAcronymList	22
\@Gls@: add \glsifplural,		compatible-3.07: new	38
\glscapscase,		\defglsdisplay: obsolete	122
\glsinsert	141	\defglsdisplayfirst: obsolete	122
change to using \glsentryfmt		\defglsentryfmt: new	74
style commands	141	\forglentries: replaced \ifx	
removed \makefirstuc (now dealt with in \glsentryfmt)	141	with \ifdefempty	67
\@Glspl@: add \glsifplural,		\gls@assign@desc: new	97
\glscapscase,		\gls@defglossaryentry: Fixed	
\glsinsert	144	default counter if none	
change to using \glsentryfmt		supplied	102
style commands	144	\gls@doentryfmt: new	74
removed \makefirstuc (now dealt with in \glsentryfmt)	144	\glsdisplay: obsolete	122
\@acrlong: added \glslabel,		\glsdisplayfirst: obsolete ..	122
\glsifplural, \glscapscase,		\glsgenentryfmt: new	117
\glsinsert and		\glsgetgrouptitle: Added check	
\glsinsert	365	in case non-Latin alphabet in	
\@acrrshort: added \glslabel,		use	244
\glsifplural, \glscapscase,		\gls glossarymark: replaced	
\glsinsert and		\MakeUppercase with	
\glsinsert	364	\mfirstucMakeUppercase ..	54
\@gls@: add \glslabel,		\glsnavigation: switched to	
\glsifplural, \glscapscase,		using \@gls@getgrouptitle	283
\glsinsert	140	\ifglshasdesc: replaced	
change to using \glsentryfmt		\ifdefempty with	
style commands	141	\ifcempty	71
\@gls@noexpand@fields: Fixed		\ifglshaslong: new	71
bug expand replaced with		\ifglshasshort: new	71
noexpand	85	\ifglshassymbol: replaced	
\@glsdisp: add \glslabel,		\ifdefempty with	
\glsifplural, \glscapscase,		\ifcempty	71
\glsinsert	145	\ifglsused: replaced	
change to using \glsentryfmt		\ifthenelse with \ifbool ..	68
style commands	145	\longnewglossaryentry: new ..	97
		\ns@newglossary: replaced	
		\glsdisplay and	
		\glsdisplayfirst with	
		\glsentryfmt	76
		numbers: new	40
		\SetCustomDisplayStyle:	
		updated to use	
		\defglsentryfmt	273

\SetDefaultAcronymDisplayStyle:	\gls@assign@desc@field:
changed to use	changed to use
\defglsentryfmt 270	\glssetnoexpandfield 24
\SetDescriptionAcronymDisplayStyle:	\gls@assign@descplural@field:
updated to use	changed to use
\defglsentryfmt 271	\glssetnoexpandfield 25
\SetDescriptionDUAAcronymDisplayStyle:	\gls@assign@name@field:
updated to use	changed to use
\defglsentryfmt 271	\glssetnoexpandfield 25
\SetDescriptionFootnoteAcronymDisplayStyle:	\gls@assign@type@field:
updated to use	changed to use
\defglsentryfmt 271	\glssetexpandfield 24
\SetDUADisplayStyle: updated	\gls@checkseeallowed: new . . 81
to use \defglsentryfmt . . 272	\glsaddallunused: set default to
\SetFootnoteAcronymDisplayStyle:	\@glo@types 182
updated to use	\Glsentryfull: changed to use
\defglsentryfmt 271	\acrfullformat 178
\SetSmallAcronymDisplayStyle:	\glsentryfull: changed to use
updated to use	\acrfullformat 178
\defglsentryfmt 272	\Glsentryfullpl: changed to use
\setupglossaries: new 45	\acrfullformat 179
\showglolong: new 277	\glsentryfullpl: changed to use
\showgloshort: new 277	\acrfullformat 178
symbols: new 40	\gls glossarymark: renamed
3.12a – 2013-10-16	\glossarymark to
\gls@defglossaryentry: added	\gls glossarymark to avoid
\glslabel 98	conflict with memoir 54
\glsaddkey: new 91	\glsprestandardsort: new . . . 15
3.13a – 2013-11-05	\glssetexpandfield: new 24
\@gls@assign@symbol@field:	\glssetnoexpandfield: new . . 24
changed to use	long: switched to
\glssetnoexpandfield 25	\tabularnewline 291, 292
\@gls@assign@symbolplural@field:	long3col: switched to
changed to use	\tabularnewline 293
\glssetnoexpandfield 25	long3colheader: switched to
\@gls@link: removed \relax . . 127	\tabularnewline 294
\@gls@notranslatorhook: new . . 29	long3colheaderborder: switched
\@gls@setupsort@standard:	to \tabularnewline 294
moved \@gls@santizesort to	long4col: switched to
\glsprestandardsort 16	\tabularnewline 294
altsuper4colheader: switched to	long4colheader: switched to
\tabularnewline 319	\tabularnewline 295
altsuper4colheaderborder:	longheader: switched to
switched to	\tabularnewline 292
\tabularnewline 320	longheaderborder: switched to
\glossarysection: changed	\tabularnewline 292
\glossarymark to	see: added
\gls glossarymark 54	\gls@checkseeallowed 80
	super: switched to
	\tabularnewline 314

super3col: switched to		\@acrfull: now using	
\tabularnewline	316	\acrfullfmt	253
super3colheader: switched to		\@gls@indexdef: new	41
\tabularnewline	317	\@gls@numbersdef: new	41
super4col: switched to		\@gls@symbolsdef: new	40
\tabularnewline	318	\ACRfullfmt: new	253
super4colheader: switched to		\Acrfullfmt: new	252
\tabularnewline	318	\acrfullfmt: new	252
super4colheaderborder:		\ACRfullplfmt: new	253
switched to		\Acrfullplfmt: new	253
\tabularnewline	319	\acrfullplfmt: new	253
superheader: switched to		\acronymentry: new	257
\tabularnewline	315	dua: new	263
superheaderborder: switched to		dua-desc: new	265
\tabularnewline	315	footnote: new	265
ucmark: added check for memoir	14	footnote-desc: new	267
3.14a – 2013-11-12		footnote-sc: new	267
General: new	338	footnote-sc-desc: new	268
\@glswritefiles: renamed		footnote-sm: new	267
\glswritefiles to		footnote-sm-desc: new	268
\@glswritefiles and used		\Genacrfullformat: new	122
“savewrites” option to set		\genacrfullformat: new	121
\glswritefiles	204	\GenericAcronymFields: new	257
acronyms: new	20	\Genplacrfullformat: new	122
compatible-2.07: added check		\genplacrfullformat: new	122
for 2.07 before setting 3.07		\Glsentryfull: bug fix: added	
compatibility	39	missing \acronymfont	178
\@gls@defglossaryentry: added		\Glsentryfull: bug fix: added	
check for existence of default		missing \acronymfont	178
glossary	99	\Glsentryfullpl: bug fix: added	
set the default for firstplural to		missing \acronymfont	179
be the value of plural	101	\Glsentryfullpl: bug fix: added	
\longprovideglossaryentry:		missing \acronymfont	178
new	98	\glsgenacfmt: new	119
notranslate: new	30	\GlsUseAcrEntryDisplayStyle:	
\provideglossaryentry: new	86	new	258
xindygloss: new	34	\GlsUseAcrStyleDefs: new	258
4.0 – 2013-11-14		index: new	41
\@gls@defglossaryentry: added		long-sc-short: new	260
check for first key	101	long-sc-short-desc: new	262
super: fixed typo in		long-short: new	258
\subglossentry		long-short-desc: new	261
(\glossentrydesc)	314	long-sm-short: new	261
4.01 – 2013-11-16		long-sm-short-desc: new	262
General: fixed non-value options		\newacronymstyle: new	258
so that they can be passed to		numberedsection: added nameref	
document class	11	option	10
4.02 – 2013-12-05		sanitize: fixed bug that caused	
General: Removed		an error here	29
\acronymfont	165–169	sc-short-long: new	261

sc-short-long-desc: new	263	moved \@gls@link@opts and	
\setacronymstyle: new	257	\@gls@link@label to	
\SetGenericNewAcronym: new	255	\@gls@link	127
short-long: new	260	\@gls@writedef: added	
short-long-desc: new	262	\glsdetoklabel	89
sm-short-long: new	261	\@glsdisp: removed \glslabel	
sm-short-long-desc: new	263	(defined in \@gls@link)	145
xindynoglsnumbers: new	34	\@glspl@: removed \glslabel	
4.02 – 2013-13-05		(defined in \@gls@link)	143
\makeglossaries: made		\@printglossary: added	
preamble only	199	\glsdetoklabel	219
4.03 – 2014-01-17		\compatibleglossentry: added	
General: changed default to		\glsdetoklabel	346
\@empty instead of \relax	38	\compatiblesubglossentry:	
4.03 – 2014-01-20		added \glsdetoklabel	346
General: removed \glslabel		dua: fixed bug in \acrfullfmt	264
(defined in \@gls@link)	161	fixed bug in \Acrfullplfmt	265
\@do@esc@wrglossary: added		fixed bug in \acrfullplfmt	265
\glsdetoklabel	212	redefined to use accessibility	
\@do@noesc@wrglossary: added		information	376
\glsdetoklabel	209	dua-desc: commented spurious	
\@ACRlong: removed \glslabel		EOL	265
(defined in \@gls@link)	366	redefined to use accessibility	
\@ACRshort: removed \glslabel		information	378
(defined in \@gls@link)	365	footnote: redefined to use	
\@Acrlong: removed \glslabel		accessibility information	379
(defined in \@gls@link)	366	footnote-desc: redefined to use	
\@Acrshort: removed \glslabel		accessibility information	381
(defined in \@gls@link)	364	footnote-sc: redefined to use	
\@GLS@: removed \glslabel		accessibility information	380
(defined in \@gls@link)	142	footnote-sc-desc: redefined to	
\@GLSpl: removed \glslabel		use accessibility information	381
(defined in \@gls@link)	144	footnote-sm: redefined to use	
\@Gls@: removed \glslabel		accessibility information	381
(defined in \@gls@link)	141	footnote-sm-desc: redefined to	
\@Gls@entry@field: new	170	use accessibility information	381
\@Glspl@: removed \glslabel		\Genacrfullformat: redefined to	
(defined in \@gls@link)	144	use accessibility information	364
\@acrlong: removed \glslabel		\genacrfullformat: redefined to	
(defined in \@gls@link)	365	use accessibility information	363
\@acrshort: removed \glslabel		\Genplacrfullformat: redefined	
(defined in \@gls@link)	364	to use accessibility	
\@gls@: removed \glslabel		information	364
(defined in \@gls@link)	140	\genplacrfullformat: redefined	
\@gls@access@display: new	355	to use accessibility	
\@gls@entry@field: new	169	information	364
\@gls@fetchfield: new	90	\glossentryname: added	
\@gls@field@link: new	146	\glsdetoklabel	241
\@gls@link: added		\gls@defglossaryentry: added	
\glsdetoklabel	127	\glsdetoklabel	98

long-sm-short-desc: redefined to use accessibility information	375	\showgloplural: added	
\renewacronymstyle: new	258	\glsdetoklabel	274
sc-short-long-desc: redefined to use accessibility information	375	\showglopluralaccess: added	
short-long: commented spurious EOL	260	\glsdetoklabel	383
short-long-desc: redefined to use accessibility information	375	\showgloshort: added	
\showglocounter: added		\glsdetoklabel	277
\showglodesc: added		\showgloshortaccess: added	
\showglodescaccess: added		\glsdetoklabel	384
\showglodescplural: added		\showgloshortpluralaccess: added	
\showglodescpluralaccess: added		\glsdetoklabel	384
\showglofirst: added		\showglosort: added	
\showglofirstaccess: added		\glsdetoklabel	277
\showglofirstpl: added		\showglosymbol: added	
\showglofirstpluralaccess: added		\glsdetoklabel	277
\showgloflag: added		\showglosymbolaccess: added	
\showgloindex: added		\glsdetoklabel	383
\showglolevel: added		\showglosymbolplural: added	
\showglolong: added		\glsdetoklabel	277
\showglolongaccess: added		\showglosymbolpluralaccess: added	
\showglolongpluralaccess: added		\glsdetoklabel	383
\showglongname: added		\showglotext: added	
\showglongnameaccess: added		\glsdetoklabel	274
\showgloparent: added		\showglotextaccess: added	
		\glsdetoklabel	383
		\showglotype: added	
		\glsdetoklabel	275
		\showglouseri: added	
		\glsdetoklabel	275
		\showglouserii: added	
		\glsdetoklabel	275
		\showglouseriii: added	
		\glsdetoklabel	276
		\showglouseriv: added	
		\glsdetoklabel	276
		\showglouserv: added	
		\glsdetoklabel	276
		\showglouservi: added	
		\glsdetoklabel	276
		sm-short-long-desc: redefined to use accessibility information	376
		4.04 – 2014-03-04	
		\@gls@getcounterprefix: added	
		warning if no prefix can be formed	213
		4.04 – 2014-03-06	
		General: added sort key to printgloss group	238
		\@@gls@noidx@nosanitizesort: new	26

<code>\@gls@noidx@sanitizesort:</code>		<code>\gls@defglossaryentry:</code>	
new	25	replaced <code>\ifx</code> with	
<code>\@gls@nosanitizesort:</code> new	25	<code>\ifdefvoid</code>	103
<code>\@gls@sanitizesort:</code> new	25	<code>\glsnoidxdisplayloc:</code> new	235
<code>\@glo@addchildren:</code> new	227	<code>\glsnoidxdisplayloclisthandler:</code>	
<code>\@glo@do@sortentries:</code> new	227	new	235
<code>\@glo@grabfirst:</code> new	232	<code>\glsnoidxloclist:</code> new	234
<code>\@glo@sortedinsert:</code> new	228	<code>\glsnoidxloclisthandler:</code> new	234
<code>\@glo@sortentries:</code> new	226	<code>\glsnoidxstripaccents:</code> new	26
<code>\@glo@sorthandler@case:</code> new	229	<code>\makeglossaries:</code> Moved	
<code>\@glo@sorthandler@letter:</code>		definition of <code>\glswrite</code> to	
new	228	<code>\makeglossaries</code>	198
<code>\@glo@sorthandler@nocase:</code>		<code>\makenoidxglossaries:</code> new	200
new	229	<code>\printglossary:</code> changed to use	
<code>\@glo@sorthandler@word:</code> new	228	new <code>\@printglossary</code>	217
<code>\@glo@sortmacro@case:</code> new	230	<code>\printnoidxglossaries:</code> new	217
<code>\@glo@sortmacro@def:</code> new	230	<code>\printnoidxglossary:</code> new	217
<code>\@glo@sortmacro@def@do:</code> new	231	<code>\showgloclist:</code> new	278
<code>\@glo@sortmacro@letter:</code> new	229	<code>\warn@noprintglossary:</code>	
<code>\@glo@sortmacro@nocase:</code> new	230	Activate warning in	
<code>\@glo@sortmacro@standard:</code>		<code>\makeglossaries</code>	216
new	230	<code>\writeist:</code> checked for definition	
<code>\@glo@sortmacro@use:</code> new	231	of <code>\glswrite</code>	184, 188
<code>\@glo@sortmacro@word:</code> new	229	4.06 – 2014-03-12	
<code>\@gls@noidx@do:</code> new	233	General: added	
<code>\@gls@noidx@getgrouptitle:</code>		<code>\glsifhyper</code>	162–169
new	244	<code>\@GLS@:</code> added <code>\glsifhyper</code>	142
<code>\@gls@noref@warn:</code> new	203	<code>\@GLSpl:</code> added <code>\glsifhyper</code>	145
<code>\@gls@reference:</code> new	235	<code>\@GLs@:</code> added <code>\glsifhyper</code>	141
<code>\@gls@warnonglossdefined:</code> new	24	<code>\@GLspl@:</code> added <code>\glsifhyper</code>	144
<code>\@gls@warnontheglossdefined:</code>		<code>\@gls@:</code> added <code>\glsifhyper</code>	141
new	24	<code>\@gls@numbersdef:</code> added hook	
<code>\@no@makeglossaries:</code> new	203	to set toc title	41
<code>\@print@glossary:</code> new	220	<code>\@gls@symbolsdef:</code> added hook	
<code>\@print@noidx@glossary:</code> new	231	to set toc title	40
<code>\@printgloss@setsort:</code> new	217	<code>\@glsdisp:</code> added <code>\glsifhyper</code>	145
<code>\@printglossary:</code> new	218	<code>\@glspl@:</code> added <code>\glsifhyper</code>	143
<code>alltree:</code> moved hangindent and		acronym: added hook to set toc	
parindent assignments outside		title	20
level test	334	acronyms: added hook to set toc	
<code>\compatibleglossentry:</code> changed		title	20
<code>\newcommand</code> to <code>\def</code> as is		<code>\glsdefmain:</code> added hook to set	
may or may not be defined	346	toc title	19
<code>\compatiblesubglossentry:</code>		4.07 – 2014-04-04	
changed <code>\newcommand</code> to <code>\def</code>		<code>\@glossarysection:</code> added	
as is may or may not be		optional argument when using	
defined	346	unstarred version	55
<code>\glo@grabfirst:</code> new	232	<code>\@gls@noidx@do:</code> added <code>\global</code>	
		in case it's used in a	
		tabular-like style	233

<code>\Acrfullplfmt</code> : fixed no case	
change bug	253
<code>\glsletentryfield</code> : new	169
4.08 – 2014-07-30	
General: added <code>entrycounter</code>	
option to <code>printgloss</code> family	237
added <code>nopostdot</code> option to	
<code>printgloss</code> family	237
added <code>subentrycounter</code> option	
to <code>printgloss</code> family	238
explicitly initialise hyper key	123
moved <code>\glsifhyper</code>	162–169
removed <code>\@sACRlongpl</code>	168
removed <code>\@sAcrlongpl</code>	168
removed <code>\@sacrlongpl</code>	167
removed <code>\@sACRlong</code>	166
removed <code>\@sAcrlong</code>	166
removed <code>\@sacrlong</code>	165
removed <code>\@sACRshortpl</code>	164
removed <code>\@sAcrshortpl</code>	164
removed <code>\@sacrshortpl</code>	163
removed <code>\@sACRshort</code>	162
removed <code>\@sAcrshort</code>	162
removed <code>\@sacrshort</code>	161
removed <code>\@gls@link</code>	125
removed <code>\@sGLSdescplural</code>	153
removed <code>\@sGLsdescplural</code>	153
removed <code>\@sglsdescplural</code>	152
removed <code>\@sGLSdesc</code>	152
removed <code>\@sGLsdesc</code>	152
removed <code>\@sglsdesc</code>	151
removed <code>\@sglsdisp</code>	145
removed <code>\@sGLSfirstplural</code>	150
removed <code>\@sGLsfirstplural</code>	150
removed <code>\@sglsfirstplural</code>	149
removed <code>\@sGLSfirst</code>	148
removed <code>\@sGLsfirst</code>	148
removed <code>\@sglsfirst</code>	147
removed <code>\@sGLSname</code>	151
removed <code>\@sGLsname</code>	151
removed <code>\@sglsname</code>	150
removed <code>\@sGLSplural</code>	149
removed <code>\@sGLsplural</code>	149
removed <code>\@sglsplural</code>	148
removed <code>\@sGLSpl</code>	144
removed <code>\@sGLspl</code>	143
removed <code>\@sglspl</code>	143
removed <code>\@sGLSsymbolplural</code>	155
removed <code>\@sGLssymbolplural</code>	155
removed <code>\@sglssymbolplural</code>	154
removed <code>\@sGLSsymbol</code>	154
removed <code>\@sGLssymbol</code>	154
removed <code>\@sglssymbol</code>	153
removed <code>\@sGLStext</code>	147
removed <code>\@sGlstext</code>	147
removed <code>\@sglstext</code>	146
removed <code>\@sGLSuseriii</code>	158
removed <code>\@sGlsuseriii</code>	158
removed <code>\@sglsuseriii</code>	157
removed <code>\@sGLSuserii</code>	157
removed <code>\@sGlsuserii</code>	157
removed <code>\@sglsuserii</code>	156
removed <code>\@sGLSuseriv</code>	159
removed <code>\@sGlsuseriv</code>	159
removed <code>\@sglsuseriv</code>	158
removed <code>\@sGLSuseri</code>	156
removed <code>\@sGlsuseri</code>	156
removed <code>\@sglsuseri</code>	155
removed <code>\@sGLSuservi</code>	161
removed <code>\@sGlsuservi</code>	161
removed <code>\@sglsuservi</code>	160
removed <code>\@sGLSuserv</code>	160
removed <code>\@sGlsuserv</code>	160
removed <code>\@sglsuserv</code>	159
removed <code>\@sGLS</code>	142
removed <code>\@sGls</code>	141
removed <code>\@sgls</code>	140
removed <code>\@thirdofthree</code>	
(defined in kernel)	140
removed <code>sPGLS</code>	344
removed <code>sPglS</code>	342
removed <code>spglS</code>	341
removed <code>sPGLSpl</code>	344
removed <code>sPglSpl</code>	343
removed <code>spglSpl</code>	342
<code>\@ACRlong</code> : added	
<code>\do@gls@link@checkfirsthyper</code>	366
<code>\@ACRshort</code> : added	
<code>\do@gls@link@checkfirsthyper</code>	365
<code>\@Acrlong</code> : added	
<code>\do@gls@link@checkfirsthyper</code>	365
<code>\@Acrshort</code> : added	
<code>\do@gls@link@checkfirsthyper</code>	364
<code>\@GLS@</code> : moved <code>\glsifhyper</code>	142
moved check for first use to	
<code>\@gls@link</code>	142

\@GLSp1: moved \glsifhyper ..	145	\ACRfullpl: removed	
moved check for first use to		\s@ACRfullpl	255
\@gls@link	145	switched to using	
\@Gls0: moved \glsifhyper ...	141	\@gls@hyp@opt	255
moved check for first use to		\Acrfullpl: removed	
\@gls@link	141	\s@Acrfullpl	254
\@Glspl0: moved \glsifhyper .	144	switched to using	
moved check for first use to		\@gls@hyp@opt	254
\@gls@link	144	\acrfullpl: removed	
\@acrlong: added		\s@acrfullpl	254
\do@gls@link@checkfirsthyper		switched to using	
.....	365	\@gls@hyp@opt	254
\@acrshort: added		\ACRlong: switched to using	
\do@gls@link@checkfirsthyper		\@gls@hyp@opt	166
.....	364	\Acrlong: switched to using	
\@closegls: new	194	\@gls@hyp@opt	166
\@gls0: moved \glsifhyper ...	141	\acrlong: switched to using	
moved check for first use to		\@gls@hyp@opt	165
\@gls@link	141	\ACRlongpl: switched to using	
\@gls@automake: new	194	\@gls@hyp@opt	168
\@gls@doautomake: new	37	\Acrlongpl: switched to using	
\@gls@field@link: added		\@gls@hyp@opt	168
assignment of		\acrlongpl: switched to using	
\do@gls@link@checkfirsthyper		\@gls@hyp@opt	167
.....	146	\ACRshort: switched to using	
\@gls@forbidtextext: new	74	\@gls@hyp@opt	162
\@gls@hyp@opt: new	124	\Acrshort: switched to using	
\@gls@link: removed		\@gls@hyp@opt	162
redundancy	127	\acrshort: switched to using	
renamed \gls@type to		\@gls@hyp@opt	161
\glstype	127	\ACRshortpl: switched to using	
\@gls@link@checkfirsthyper:		\@gls@hyp@opt	164
new	126	\Acrshortpl: switched to using	
\@glsdisp: moved \glsifhyper	145	\@gls@hyp@opt	164
moved check for first use to		\acrshortpl: switched to using	
\@gls@link	145	\@gls@hyp@opt	163
\@glspl0: moved \glsifhyper .	143	altlongragged4col: fixed bug	
moved check for first use to		that displayed description	
\@gls@link	143	instead of symbol	306
\@ignored@glossaries: new ..	78	automake: new	37
\ACRfull: removed \s@ACRfull	254	\forallacronyms: new	67
switched to using		\GLS: switched to using	
\@gls@hyp@opt	254	\@gls@hyp@opt	142
\Acrfull: removed \s@Acrfull	253	\Gls: switched to using	
switched to using		\@gls@hyp@opt	141
\@gls@hyp@opt	253	\gls: switched to using	
\acrfull: removed \s@acrfull	253	\@gls@hyp@opt	140
switched to using		\gls@defglossaryentry: added	
\@gls@hyp@opt	253	check for ignored glossary .	100
		\gls@istfilebase: new	51

\glsaddkey: removed		\glsname: switched to using	
\@sGLS@user@ <i>key</i>)	93	\@gls@hyp@opt	150
removed \@sGls@user@ <i>key</i>)	93	\GLSpl: switched to using	
removed \@sgls@user@ <i>key</i>)	92	\@gls@hyp@opt	144
switched to using		\Glspl: switched to using	
\@gls@hyp@opt	92, 93	\@gls@hyp@opt	143
\GLSdesc: switched to using		\glspl: switched to using	
\@gls@hyp@opt	152	\@gls@hyp@opt	142
\Glsdesc: switched to using		\GLSplural: switched to using	
\@gls@hyp@opt	151	\@gls@hyp@opt	149
\glsdesc: switched to using		\Glsplural: switched to using	
\@gls@hyp@opt	151	\@gls@hyp@opt	148
\GLSdescplural: switched to		\glsplural: switched to using	
using \@gls@hyp@opt	153	\@gls@hyp@opt	148
\Glsdescplural: switched to		\glspace: new	252
using \@gls@hyp@opt	152	\GLSsymbol: switched to using	
\glsdescplural: switched to		\@gls@hyp@opt	154
using \@gls@hyp@opt	152	\Glsymbol: switched to using	
\glsdisablehyper: added		\@gls@hyp@opt	153
\KV@glslink@hyperfalse to		\glsymbol: switched to using	
definition	139	\@gls@hyp@opt	153
\glsdisp: switched to using		\GLSsymbolplural: switched to	
\@gls@hyp@opt	145	using \@gls@hyp@opt	155
\glsdohyperlink: new	139	\Glsymbolplural: switched to	
\glsdohypertarget: new	137	using \@gls@hyp@opt	154
\glsenablehyper: added		\glsymbolplural: switched to	
\KV@glslink@hypertrue to		using \@gls@hyp@opt	154
definition	140	\GLStext: switched to using	
\GLSfirst: switched to using		\@gls@hyp@opt	146
\@gls@hyp@opt	148	\Glstext: switched to using	
\Glsfirst: switched to using		\@gls@hyp@opt	147
\@gls@hyp@opt	147	\glstext: switched to using	
\glsfirst: switched to using		\@gls@hyp@opt	146
\@gls@hyp@opt	147	\glstreenamefmt: new	327
\GLSfirstplural: switched to		\GLSuseri: switched to using	
using \@gls@hyp@opt	150	\@gls@hyp@opt	156
\Glsfirstplural: switched to		\Glsuseri: switched to using	
using \@gls@hyp@opt	149	\@gls@hyp@opt	155
\glsfirstplural: switched to		\glsuseri: switched to using	
using \@gls@hyp@opt	149	\@gls@hyp@opt	155
\glsifhyper: deprecated	124	\GLSuserii: switched to using	
\glslink: switched to using		\@gls@hyp@opt	157
\@gls@hyp@opt	125	\Glsuserii: switched to using	
\glslinkcheckfirsthyperhook:		\@gls@hyp@opt	156
new	126	\glsuserii: switched to using	
\glslinkvar: new	124	\@gls@hyp@opt	156
\GLSname: switched to using		\GLSuseriii: switched to using	
\@gls@hyp@opt	151	\@gls@hyp@opt	158
\Glsname: switched to using		\Glsuseriii: switched to using	
\@gls@hyp@opt	150	\@gls@hyp@opt	157

<code>\glsuseriii</code> : switched to using		<code>\@gls@hyp@opt</code>	157	<code>\@gls@glossary</code> : Renamed	
<code>\GLSuseriv</code> : switched to using		<code>\@gls@hyp@opt</code>	159	<code>\@glossary</code> to	
<code>\Glsuseriv</code> : switched to using		<code>\@gls@hyp@opt</code>	158	<code>\@gls@glossary</code>	205
<code>\glsuseriv</code> : switched to using		<code>\@gls@hyp@opt</code>	158	<code>alttree</code> : moved space after	
<code>\GLSuseriv</code> : switched to using		<code>\@gls@hyp@opt</code>	158	symbol	334, 335
<code>\Glsuseriv</code> : switched to using		<code>\@gls@hyp@opt</code>	158	<code>\glspercentchar</code> : new	183
<code>\GLSuserv</code> : switched to using		<code>\@gls@hyp@opt</code>	160	<code>\glstildechar</code> : new	183
<code>\Glsuserv</code> : switched to using		<code>\@gls@hyp@opt</code>	159	4.11 – 2014-09-01	
<code>\glsuserv</code> : switched to using		<code>\@gls@hyp@opt</code>	159	<code>\@do@esc@wrglossary</code> : added	
<code>\GLSuservi</code> : switched to using		<code>\@gls@hyp@opt</code>	161	hook	211
<code>\Glsuservi</code> : switched to using		<code>\@gls@hyp@opt</code>	160	<code>\gls@wrglossary</code> : renamed from	
<code>\glsuservi</code> : switched to using		<code>\@gls@hyp@opt</code>	160	<code>\@wrglossary</code> to	
<code>\ifignoredglossary</code> : new	78	<code>\@gls@hyp@opt</code>	160	<code>\gls@wrglossary</code>	206
<code>\newglossary</code> : added starred		<code>\@gls@hyp@opt</code>	160	<code>\glsaddprotectedpagefmt</code> : new	208
version	75	<code>\@gls@hyp@opt</code>	160	<code>\glsbackslash</code> : new	182
<code>\newignoredglossary</code> : new	77	<code>\@gls@hyp@opt</code>	160	<code>sanitize</code> : none option	29
<code>\ns@newglossary</code> : added		<code>\@gls@hyp@opt</code>	160	4.12 – 2014-11-22	
<code>\@glotype@{<i>name</i>}@log</code>	76	<code>\@gls@hyp@opt</code>	160	<code>\@gls@addpredefinedattributes</code> :	
new	75	<code>\@gls@hyp@opt</code>	160	Added <code>glsignore</code> attribute	61
<code>\p@gls@hyp@opt</code> : new	125	<code>\@gls@hyp@opt</code>	160	<code>\@gls@adjustmode</code> : new	181
<code>\PGLS</code> : changed to use		<code>\@gls@hyp@opt</code>	160	<code>\@gls@notranslatorhook</code> :	
<code>\@gls@hyp@opt</code>	344	<code>\@gls@hyp@opt</code>	160	removed	29
<code>\PglS</code> : changed to use		<code>\@gls@hyp@opt</code>	160	<code>\@gls@toc</code> : added <code>\protect</code> to	
<code>\@gls@hyp@opt</code>	342	<code>\@gls@hyp@opt</code>	160	<code>\numberline</code>	57
<code>\pglS</code> : changed to use		<code>\@gls@hyp@opt</code>	160	<code>\@gls@usetranslator</code> : new	29
<code>\@gls@hyp@opt</code>	341	<code>\@gls@hyp@opt</code>	160	<code>\glsacrpluralsuffix</code> : new	48
<code>\PGLSpl</code> : changed to use		<code>\@gls@hyp@opt</code>	160	<code>\glsadd</code> : added check for vertical	
<code>\@gls@hyp@opt</code>	344	<code>\@gls@hyp@opt</code>	160	mode	181
<code>\PglSpl</code> : changed to use		<code>\@gls@hyp@opt</code>	160	<code>\glsaddallunused</code> : replaced	
<code>\@gls@hyp@opt</code>	343	<code>\@gls@hyp@opt</code>	160	<code>@gobble</code> with <code>glsignore</code>	182
<code>\pglSpl</code> : changed to use		<code>\@gls@hyp@opt</code>	160	<code>\glsifusedtranslatordict</code> : new	29
<code>\@gls@hyp@opt</code>	342	<code>\@gls@hyp@opt</code>	160	<code>\glsignore</code> : new	182
<code>\s@gls@hyp@opt</code> : new	124	<code>\@gls@hyp@opt</code>	160	<code>\glsupacrpluralsuffix</code> : new	48
<code>\s@newglossary</code> : new	75	<code>\@gls@hyp@opt</code>	160	<code>\ProvidesGlossariesLang</code> : new	48
4.09 – 2014-08-12		<code>\@gls@hyp@opt</code>	160	<code>\RequireGlossariesLang</code> : new	48
<code>\glsaddkey</code> : fixed bug in user		<code>\@gls@hyp@opt</code>	160	4.13 – 2015-02-03	
commands	92	<code>\@gls@hyp@opt</code>	160	<code>\indexspace</code> : new	287, 308, 327
4.10 – 2014-08-27		<code>\@gls@hyp@opt</code>	160	4.14 – 2015-02-28	
<code>\@Gls@acrentryname</code> : new	171	<code>\@gls@hyp@opt</code>	160	<code>\@@glslocalreset</code> : new	110
<code>\@Gls@entryname</code> : new	171	<code>\@gls@hyp@opt</code>	160	<code>\@@glslocalunset</code> : new	109
		<code>\@gls@hyp@opt</code>	160	<code>\@@glsreset</code> : new	110
		<code>\@gls@hyp@opt</code>	160	<code>\@@glsunset</code> : new	110
		<code>\@gls@hyp@opt</code>	160	<code>\@@newglossaryentry@defcounters</code> :	
		<code>\@gls@hyp@opt</code>	160	new	111
		<code>\@gls@hyp@opt</code>	160	<code>\@cGls</code> : new	115
		<code>\@gls@hyp@opt</code>	160	<code>\@cGls@</code> : new	115
		<code>\@gls@hyp@opt</code>	160	<code>\@cGlspl@</code> : new	116
		<code>\@gls@hyp@opt</code>	160	<code>\@cgls</code> : new	114
		<code>\@gls@hyp@opt</code>	160	<code>\@cgls@</code> : new	114
		<code>\@gls@hyp@opt</code>	160	<code>\@cglspl</code> : new	115, 116

\cglspl@: new	115	\@GLSpl: added	
\@gls@entry@count: new	114	\glspostlinkhook	145
\@gls@increment@currcount:		\@Gls@: added	
new	113	\glspostlinkhook	142
\@gls@local@increment@currcount:		\@GLspl@: added	
new	113	\glspostlinkhook	144
\@gls@write@entrycounts: new	114	\@acrlong: added	
\@glslocalreset: new	110	\glspostlinkhook	365
\@glslocalunset: new	109	\@acrshort: added	
\@glsreset: new	110	\glspostlinkhook	364
\@glsunset: new	109	\@gls@: added	
\@newglossaryentry@defcounters:		\glspostlinkhook	141
new	104	\@gls@@link: added	
\cGls: new	115	\glspostlinkhook	125
\cgl: new	114	\@gls@field@link: added	
\cGlsformat: new	115	\glspostlinkhook	146
\cglformat: new	114	\@gls@link: moved definition of	
\cGLspl: new	116	\glsifhyperon outside of this	
\cglspl: new	115	macro	127
\cGlsplformat: new	116	\@glsdisp: added	
\cglsplformat: new	116	\glspostlinkhook	145
\gls@defdocnewglossaryentry:		\@GLspl@: added	
new	86	\glspostlinkhook	143
\glsenableentrycount: new	111	\glsacspace: new	260
\glslocalreset: switched to		\glsadd: changed	
\@glslocalreset	109	\@do@wrglossary to	
\glslocalunset: switched to		\@do@wrglossary	181
\@glslocalunset	109	\glsfielddef: new	95
\glsreset: switched to		\glsfieldedef: new	94
\@glsreset	108	\glsfieldfetch: new	95
\glsunset: switched to		\glsfieldgdef: new	95
\@glsunset	109	\glsfieldxdef: new	94
4.15 – 2015-03-16		\glsifhyperon: moved definition	
General: bug fix replaced		of \glsifhyperon	126
\@glo@type with \glstype	168	\glslinkpostsetkeys: new	126
4.16 – 2015-06-18		\glspostlinkhook: new	126
\glsaddstoragekey: new	91	\glswriteentry: new	206
4.16 – 2015-07-08		\ifglsfieldcseq: new	97
General: added		\ifglsfielddefeq: new	96
\glspostlinkhook	162–169	\ifglsfieldeq: new	96
\@ACRlong: added		long-sp-short: new	259
\glspostlinkhook	366	long-sp-short-desc: new	262
\@ACRshort: added		\showglofield: new	278
\glspostlinkhook	365	4.18 – 2015-09-09	
\@Acrlong: added		General: split mfirstuc into	
\glspostlinkhook	366	separate bundle	4
\@Acrshort: added		4.19 – 2015-10-31	
\glspostlinkhook	365	\glstreenamebox: new	333
\@GLS@: added			
\glspostlinkhook	142		

4.19 – 2015-11-22			
\@gls@link@nocheckfirsthyper:		\glsentrytitlecase: new	175
new	146	\glsfindwidesttoplevelname:	
\@gls@preglossaryhook: new . .	218	new	333
\@printglossary: added		\glslistgroupheaderfmt: new . .	287
\@gls@preglossaryhook . .	219	\glslistnavigationitem: new . .	287
\do@glsdisablehyperinlist:		\glstreegroupheaderfmt: new . .	327
new	126	\glstreenavigationfmt: new . .	327
\doifglossarynoexistsordo:		\ifglswrallowprimitivemods:	
new	70	new	209
\gls@gobbleopt: new	75	list: fixed missing space before	
\glsdoifexistsordo: new	69	description	288
4.20 – 2015-11-30		long: fixed typo in	
\@gls@link: added		\glossentrydesc	291
\@gls@setdefault@glslink@opts		mcolalttreespannav: new	313
.	127	mcolindexspannav: new	309
added \glsdonohyperlink		mcoltreenonamespannav: new . .	311
when hyperlink is suppressed	127	mcoltreespannav: new	310
\@gls@setdefault@glslink@opts:		super4col: fixed bug in	
new	126	\glossentry	318
\gls@checkseeallowed@preambleonly:		4.23 – 2016-04-30	
new	81	altlongragged4col: check for	
\glsdonohyperlink: new	139	nogroupskip changed	306
4.21 – 2016-01-24		altsuperragged4col: check for	
General: changed checkfirsthyper		nogroupskip changed	325
assignment	161–169	\glscurrentfieldvalue: new . .	73
\@printglossary: warn if no		\ifglshasfield: added	
style has been set	218	\glscurrentfieldvalue	72, 73
altlong4col-booktabs: new . .	299	long: check for nogroupskip	
altlongragged4col-booktabs:		changed	292
new	300	long-booktabs: check for	
\glsLTpenaltycheck: new	301	nogroupskip changed	298
\glspatchLtoutput: new	301	long3col: check for nogroupskip	
\glspenaltygroupskip: new . .	301	changed	293
long-booktabs: new	297	long3col-booktabs: check for	
long3col-booktabs: new	298	nogroupskip changed	298
long4col-booktabs: new	298	long4col: check for nogroupskip	
longragged-booktabs: new . . .	299	changed	295
longragged3col-booktabs: new .	300	long4col-booktabs: check for	
\setglossarystyle: set default		nogroupskip changed	299
style if not already set	246	longragged: check for	
4.22 – 2016-04-19		nogroupskip changed	303
\@@do@esc@wrglossary: added		longragged3col: check for	
check for \@arabic	211	nogroupskip changed	304
added test to allow temporary		super: check for nogroupskip	
primitive modifications and		changed	314
added arabic case	211	super3col: check for nogroupskip	
\gls@arabicpage: new	207	changed	316
\gls@protected@pagefmts:		super4col: check for nogroupskip	
added arabic to list	207	changed	318

superragged: check for nogroupskip changed	322	outside of theglossary environment	232
superragged3col: check for nogroupskip changed	324	\glsnavhyperlinkname: new	281
4.24 – 2016-05-27		4.30 – 2017-06-11	
\@gls@extramakeindexopts: new	193	\@glo@autosee: new	103
\@gls@glossary: added check for debug mode	205	\@glo@autoseehook: new	104
\@gls@see@noindex: new	9	\@glo@check@sortallowed: new	16
debug: new	7	\@gls@noidx@do: letter group assignment made global	234
\glsnomakeindexwarning: new	57	\@gls@setupsort@def: added check for register	17
\GlsSetQuote: new	190	\@gls@setupsort@none: new	18
\GlsSetWriteIstHook: new	190	\@xdycrossrefhook: new	63
seenindex: new	9	\@xdylocationclassorder: bug fix: changed \edef to \def	63
4.25 – 2016-06-09		\glosortentrieswarning: new	23
General: changed		\gls@set@xr@key: new	80
\DeclareRobustCommand to \newrobustcmd and changed \ifundefined to \ifcsundef	358	\gls@xr@key: new	81
\@gls@enablesavenonumberlist: new	82	\GlsAddXdyLocation: bug fix: changed #1 to #2	63
\@gls@initnonumberlist: new	82	\glsnoidxstripaccents: added \a	26
\@gls@savenonumberlist: new	82	added \TH, \dh and \DH	27
4.26 – 2016-10-12		4.31 – 2017-08-10	
\@glossary@default@style: added check for classicthesis	11	nolist: added check for “list” style	13
\glstreechildpredesc: new	328	4.31 – 2017-09-10	
\glstreeitem: new	328	style: changed \renewcommand to \def	11
\glstreepredesc: new	328	4.32 – 2017-08-24	
\glstreesubitem: new	328	\@glsnavhypertarget: new	282
\glstreesubsubitem: new	328	\@glsshowtarget: new	8
mcolindex: replaced \@idxitem with \glstreeitem	308	\glsshowtarget: new	8
mcolindexspannav: replaced \@idxitem with \glstreeitem	309	4.33 – 2017-09-20	
4.28 – 2017-01-07		\@do@esc@wrglossary: added \gls@the and \gls@number	211
\glspatchtabularx: new	108	renamed from \@do@esc@wrglossary	209
4.29 – 2017-01-19		\@do@noesc@wrglossary: new	209
General: added check for \@gls@xr@doaccsupp	346	\@do@wrglossary: changed to check for eslocations	208
\@gls@noidx@do: current letter group assignment made global	234	\@gls@missinglang@warn: new	23
\@print@noidx@glossary: moved definition of \@gls@currentlettergroup		eslocations: new	12
		\GlsSetXdyFirstLetterAfterDigits: added starred version	183
		\GlsSetXdyNumberGroupOrder: new	183

4.34 – 2017-11-03		
<code>\glsshowtarget</code> : modified to check for math mode and inner	8	
<code>mcolalttreesspannav</code> : removed spurious space	313	
4.35 – 2017-11-14		
<code>\glsadd</code> : added <code>\@gls@setsort</code> (in case of <code>sort=use</code>)	181	
4.36 – 2018-03-07		
<code>\@gls@glossary</code> : removed <code>\index</code>	205	
4.37 – 2018-04-07		
<code>\gls@begindocdefs</code> : new	88	
4.38 – 2018-05-10		
<code>\@gls@define@glossaryentrycounter</code> : added check for existence of glossaryentry counter	14	
new	14	
<code>\@gls@define@glossarysubentrycounter</code> : new	15	
prepended <code>\currentglossary</code> . to <code>\theHglossarysubentry</code> and removed spurious eol space	15	
<code>\glsaccsupp</code> : added braces around actual text argument	355	
<code>\glsentrycounterlabel</code> : bug fix: move conditional inside command	240	
<code>\GlsEntryCounterLabelPrefix</code> : new	237	
<code>\glsentryitem</code> : bug fix: move conditional inside command	240	
<code>\glsrefentry</code> : bug fix: move conditional inside command	240	
<code>\glsresetentrycounter</code> : bug fix: move conditional inside command	239	
<code>\glsresetsubentrycounter</code> : bug fix: move conditional inside command	239	
<code>\glsstepentry</code> : bug fix: move conditional inside command	239	
<code>\glsstepsubentry</code> : bug fix: move conditional inside command	239	
<code>\glssubentrycounterlabel</code> : bug fix: move conditional inside command	240	
		<code>\glssubentryitem</code> : bug fix: move conditional inside command
		<code>\showglnonameaccess</code> : bug fix: corrected field (was showing text access field)
		4.40 – 2018-06-01
		<code>\istfile</code> : changed <code>\def</code> to <code>\providecommand</code>
		<code>\makenoidxglossaries</code> : false
		4.41 – 2018-07-23
		General: changed <code>\val</code> and <code>\nr</code> to <code>\gls@numberedsection@val</code> and <code>\gls@numberedsection@nr</code>
		<code>\@gls@override@glossary</code> : new
		debug: changed <code>\val</code> and <code>\nr</code> to <code>\gls@debug@val</code> and <code>\gls@debug@nr</code>
		<code>\glossary</code> : added warning
		<code>\gls@original@glossary</code> : new
		<code>\gls@original@makeglossary</code> : new
		<code>kernelglossredefs</code> : new
		<code>\makeglossaries</code> : removed redefinition of <code>\makeglossary</code>
		<code>\makeglossary</code> : added warning
		<code>nonumberlist</code> : changed <code>\val</code> and <code>\nr</code> to <code>\gls@nonumberlist@val</code> and <code>\gls@nonumberlist@nr</code>
		<code>numberedsection</code> : changed <code>\val</code> and <code>\nr</code> to <code>\gls@numberedsection@val</code> and <code>\gls@numberedsection@nr</code>
		<code>seenoindex</code> : changed <code>\val</code> and <code>\nr</code> to <code>\gls@seenoindex@val</code> and <code>\gls@seenoindex@nr</code>
		<code>translate</code> : changed <code>\val</code> and <code>\nr</code> to <code>\gls@translate@val</code> and <code>\gls@translate@nr</code>
		4.42 – 2019-01-06
		<code>\@gls@@automake@immediate</code> : new
		<code>\@gls@automake@immediate</code> : new
		<code>altlong4col-booktabs</code> : removed superfluous <code>\glspatchLToutput</code>

automake: switch from boolean to choice	37	\@GLSuserv@: added accessibility support	371
\gls@automake@nr: new	36	\@GLSuserii@: added accessibility support	371
\glsfielddedef: changed from \edef to \protected@csedef	94	\@GLSdesc@: added accessibility support	368
\glsfieldxdef: changed from \edef to \protected@csxdef	94	\@GLSdescplural@: added accessibility support	368
\ifglsautomake: now defined explicitly instead of through boolean key	36	\@GLSfirst@: added accessibility support	367
noglossaryindex: new	41	\@GLSfirstplural@: added accessibility support	367
4.43 – 2019-09-28		\@GLSname@: added accessibility support	368
\glsnoidxstripaccents: add check for LaTeX version 2019/10/01	27	\@GLSplural@: added accessibility support	367
4.44 – 2019-12-06		\@GLssymbol@: added accessibility support	369
\glsprefix@record@hook: new	341	\@GLssymbolplural@: added accessibility support	369
4.45 – 2020-02-13		\@GLStext@: added accessibility support	366
General: removed backward compatibility use of symbol key	349	\@GLSuseri@: added accessibility support	369
\@do@write@glslabels: new	35	\@GLSuseriii@: added accessibility support	370
\@glsshowtarget: new	8	\@GLSuseriv@: added accessibility support	370
\@GLSdesc@: added accessibility support	368	\@GLSuserv@: added accessibility support	371
\@GLSdescplural@: added accessibility support	368	\@GLSuservi@: added accessibility support	371
\@GLSfirst@: added accessibility support	367	\@do@write@glslabels: new	35
\@GLSfirstplural@: added accessibility support	367	\@domakeglossaries: new	34
\@GLSname@: added accessibility support	368	\@gls@access@display: deprecated	355
\@GLSplural@: added accessibility support	367	\@gls@fieldaccess@display: new	355
\@GLSsymbol@: added accessibility support	369	\@glsdesc@: added accessibility support	368
\@GLSsymbolplural@: added accessibility support	369	\@glsdescplural@: added accessibility support	368
\@GLStext@: added accessibility support	366	\@glsfirst@: added accessibility support	367
\@GLSuseri@: added accessibility support	369	\@glsfirstplural@: added accessibility support	367
\@GLSuserii@: added accessibility support	370	\@glsname@: added accessibility support	368
\@GLSuseriii@: added accessibility support	370		
\@GLSuseriv@: added accessibility support	370		

\@glsplural@: added accessibility support	367	\Glsentryprefixfirstplural: added \glsdetoklabel	340
\@glsshowaccsupp: new	9	\glsentryprefixfirstplural: added \glsdetoklabel	339
\@glssymbol@: added accessibility support	369	\Glsentryprefixplural: added \glsdetoklabel	340
\@glssymbolplural@: added accessibility support	369	\glsentryprefixplural: added \glsdetoklabel	339
\@glstext@: added accessibility support	366	\glsentrytitlecase: added existence check	175
\@glsuseri@: added accessibility support	369	\glsentryuseriaccess: new	353
\@glsuserii@: added accessibility support	370	\glsentryuseriiaccess: new	353
\@glsuseriii@: added accessibility support	370	\glsentryuseriiiaccess: new	353
\@glsuseriv@: added accessibility support	370	\glsentryuserivaccess: new	353
\@glsuserv@: added accessibility support	371	\glsentryuserviaccess: new	353
\@glsuservi@: added accessibility support	371	\glsfieldaccsupp: new	354
\@org@Gls@acrentryname: added check for \glsshortaccessdisplay	172	\glsfirstaccessdisplay: added check for existence	356
acronyms: changed \renewcommand to \def	20	\glsfirstpluralaccessdisplay: added check for existence	356
debug: showaccsupp	7	\glslongaccessdisplay: added check for existence	357
disablemakegloss: new	34	\glslongpluralaccessdisplay: added check for existence	357
\gls@accessibility: new	346	\glsnameaccessdisplay: added check for existence	355
\gls@accsupp@engine: new	346	\glspluralaccessdisplay: added check for existence	356
\glsaccessibility: new	346	\glsprefixsep: new	341
\glsdefaultshortaccess: new	382	\glssee: switched to \newrobustcmd	214
\glsdescriptionaccessdisplay: added check for existence	356	\glsseeformat: switched to \newrobustcmd	215
\glsdescriptionpluralaccessdisplay: added check for existence	357	\glsseeitem: switched to \newrobustcmd	215
\glsentrydescpluralaccess: corrected field reference	352	\glsseelist: switched to \newrobustcmd	215
\glsentryfirstpluralaccess: switched to using \@gls@entry@field	352	\glsshortaccessdisplay: added check for existence	357
\glsentryparent: new	175	\glsshortaccsupp: new	355
\Glsentryprefix: added \glsdetoklabel	340	\glsshortplaccsupp: new	355
\glsentryprefix: added \glsdetoklabel	339	\glsshortpluralaccessdisplay: added check for existence	357
\Glsentryprefixfirst: added \glsdetoklabel	339	\glsshowaccsupp: new	9
\glsentryprefixfirst: added \glsdetoklabel	339	\glsshowtargetfont: new	8
		\glsshowtargetouter: new	8
		\glsshowtargetsymbols: new	8
		\glsymbolaccessdisplay: added check for existence	356

<code>\glssymbolpluralaccessdisplay:</code>	<code>\ifglossaryexists:</code> added	
added check for existence	starred form	67
<code>\glstextaccessdisplay:</code> added	<code>\s@ifglossaryexists:</code> new	68
check for existence	<code>\setglossarypreamble:</code> switched	
<code>\glsuseriaccessdisplay:</code> new	to starred form of	
<code>\glsuseriiaccessdisplay:</code> new	<code>\ifglossaryexists</code>	54
<code>\glsuseriiiaccessdisplay:</code>	4.47 – 2021-09-20	
new	General: changed <code>\x</code> to <code>\@gls@x</code>	248
<code>\glsuserivaccessdisplay:</code> new	<code>\@do@esc@wrglossary:</code> changed	
<code>\glsuservaccessdisplay:</code> new	<code>\edef</code> to <code>\protected@edef</code>	212
<code>\glsuserviiaccessdisplay:</code> new	<code>\@do@noesc@wrglossary:</code>	
<code>\ifglshaschildren:</code> made robust	changed <code>\edef</code> to	
<code>\ifglshasfield:</code> made robust	<code>\protected@edef</code>	209
<code>\ifglshaslong:</code> made robust	<code>\@do@write@glslabels:</code> fully	
<code>\ifglshasprefix:</code> added	expand list	35
<code>\glsdetoklabel</code>	<code>\@gls@@default@acr@descplural:</code>	
<code>\ifglshasprefixfirst:</code> added	new	270
<code>\glsdetoklabel</code>	<code>\@gls@@default@acr@firstplural:</code>	
<code>\ifglshasprefixfirstplural:</code>	new	270
added <code>\glsdetoklabel</code>	<code>\@gls@@default@acr@plural:</code>	
<code>\ifglshasprefixplural:</code> added	new	270
<code>\glsdetoklabel</code>	<code>\@gls@noexpand@assign@default@plurals:</code>	
<code>\ifglshasshort:</code> made robust	new	270
<code>\ifglshassymbol:</code> made robust	<code>\@glo@do@sortentries:</code> changed	
<code>\makeglossaries:</code> let	<code>\edef</code> to <code>\protected@edef</code>	227
<code>\@makeglossary</code> to <code>\@gobble</code>	<code>\@glo@sortentries:</code> changed	
instead of <code>\relax</code>	<code>\edef</code> to <code>\protected@edef</code>	226
<code>restoremakeloss:</code> new	<code>\@glo@storeentry:</code> changed	
<code>user1access:</code> new	<code>\edef</code> to <code>\protected@edef</code>	105
<code>user2access:</code> new	<code>\@gls@checkesc:</code> new	136
<code>user3access:</code> new	<code>\@gls@deprecated@acr@warning:</code>	
<code>user4access:</code> new	new	272
<code>user5access:</code> new	<code>\@gls@link:</code> changed <code>\edef</code> to	
<code>user6access:</code> new	<code>\protected@edef</code>	127
<code>writeglslabels:</code> new	<code>\@gls@setupsort@use:</code> changed	
<code>\xglfieldaccsupp:</code> new	<code>\edef</code> to <code>\protected@edef</code>	18
4.46 – 2020-03-19	<code>\@gls@writedef:</code> added a	
<code>\@ifglossaryexists:</code> new	warning if entry has already	
<code>\@printgloss@checkexists:</code>	been defined	89
new	<code>\@makeglossaries@warn@noprintglossary:</code>	
<code>\@printgloss@checkexists@allowignored:</code>	new	199
new	<code>\@write@glslabelsname:</code> new	36
<code>\@printgloss@checkexists@noignored:</code>	<code>\do@write@glslabelsname:</code> new	36
new	<code>\forglslentries:</code> changed <code>\edef</code>	
<code>\@printglossary:</code> replaced	to <code>\protected@edef</code>	67
<code>\ifglossaryexists</code> with	<code>\gls@defdocnewglossaryentry:</code>	
<code>\@printgloss@checkexists</code>	removed see warning	86
<code>\doifglossarynoexistsordo:</code>	<code>\gls@defglossaryentry:</code> changed	
switched to starred form of	<code>\edef</code> to	
<code>\ifglossaryexists</code>	<code>\protected@edef</code>	98, 99

<code>\gls@docdef@addlabel</code> : new ..	87	<code>\SetDescriptionFootnoteAcronymStyle</code> :	
<code>\gls@docdef@labellist</code> : new ..	87	added deprecated warnings	271
<code>\gls@glsdefs@addlabel</code> : new ..	87	<code>\SetDUASStyle</code> : added deprecated	
<code>\gls@glsdefs@labellist</code> : new ..	87	warnings	272
<code>\gls@save@numberlist</code> : changed		<code>\SetFootnoteAcronymStyle</code> :	
<code>\edef</code> to <code>\protected@edef</code>	216	added deprecated warnings	272
<code>\glsadd</code> : changed <code>\edef</code> to		<code>\SetSmallAcronymStyle</code> : added	
<code>\protected@edef</code>	181	deprecated warnings	272
<code>\glsaddall</code> : changed <code>\edef</code> to		<code>writeglslabelnames</code> : new	36
<code>\protected@edef</code>	181	4.48 – 2021-10-19	
<code>\glsdefs@newdocentry</code> : new ..	88	<code>\glscapitalisewords</code> : new	175
<code>\glsdisplaynumberlist</code> : changed		<code>\glslistexpandedname</code> : new	287
<code>\edef</code> to <code>\protected@edef</code>	179	<code>\glslistinit</code> : new	287
<code>\glsfielddef</code> : changed <code>\edef</code> to		<code>\glsunexpandedfieldvalue</code> : new	96
<code>\protected@edef</code>	95	4.49 – 2021-11-01	
<code>\glsfielddedef</code> : changed <code>\edef</code> to		<code>\glslistinit</code> : added	
<code>\protected@edef</code>	94	<code>\glsnavigation</code>	287
<code>\glsfieldfetch</code> : changed <code>\edef</code>		4.50 – 2022-10-14	
to <code>\protected@edef</code>	95	General: added initialisation	251
<code>\glsfieldgdef</code> : changed <code>\edef</code> to		allow markup to be stripped	
<code>\protected@edef</code>	95	from location	248
<code>\glsfieldxdef</code> : changed <code>\edef</code> to		corrected description mapping	349
<code>\protected@edef</code>	94	don't load <code>glossary-super</code> if <code>xtab</code>	
<code>\glsIfListOfAcronyms</code> : changed		has been loaded	12
<code>\edef</code> to <code>\protected@edef</code>	21	No longer requires <code>textcase</code>	
<code>\glsmoveentry</code> : changed <code>\edef</code> to		with <code>mfirstuc 2.08+</code>	5
<code>\protected@edef</code>	104	removed automatic loading of	
<code>\glsnavhyperlink</code> : changed		<code>glossaries-compatible-307</code>	42
<code>\edef</code> to <code>\protected@edef</code>	280	switched to using	
<code>\GlsSetQuote</code> : added check for		<code>\glsuppercase</code>	
backslash	192 147–161, 163, 165, 167, 169	
<code>\glsstepsubentry</code> : changed		<code>\@@do@esc@wrglossary</code> : moved	
<code>\edef</code> to <code>\protected@edef</code>	240	<code>\romannumeral</code> inside allow	
<code>\ifglsfieldcseq</code> : changed <code>\edef</code>		primitive mods	211
to <code>\protected@edef</code>	97	<code>\@@gls@@default@acr@descplural</code> :	
<code>\ifglshaschildren</code> : changed		removed	270
<code>\edef</code> to <code>\protected@edef</code>	70	<code>\@@gls@@default@acr@firstplural</code> :	
<code>\ifignoredglossary</code> : changed		removed	270
<code>\edef</code> to <code>\protected@edef</code>	78	<code>\@@gls@@default@acr@plural</code> :	
<code>\makenoidxglossaries</code> : changed		removed	270
<code>\edef</code> to <code>\protected@edef</code>	200	<code>\@@gls@default@entryfmt</code> :	
<code>\new@glossaryentry</code> : allow		removed	358
overwrite if defined in		<code>\@@gls@navhypertarget</code> : new	282
<code>.glsdefs</code>	87	<code>\@@gls@noexpand@assign@default@plurals</code> :	
<code>\newignoredglossary</code> : changed		removed	270
<code>\edef</code> to <code>\protected@edef</code>	77	<code>\@Gls@acronymname</code> : removed	
<code>\ns@newglossary</code> : changed <code>\edef</code>		<code>mfirstuc</code> internal commands	171
to <code>\protected@edef</code>	76	<code>\@Glsdisp</code> : new	146
<code>\SetDescriptionDUAAcronymStyle</code> :		<code>\@Glsentryfield</code> : new	170
added deprecated warning	271		

\@domakeglossaries: added		\CustomNewAcronymDef:	
check for \nofiles	34	removed	273
\@gls@automake: added check for		\DefaultNewAcronymDef:	
void codepage	195	removed	270, 382
\@gls@automake@immediate:		\defglsdisplay: removed	122
added check for void		\defglsdisplayfirst: removed	122
codepage	196	\DescriptionDUANewAcronymDef:	
\@gls@automake@types: new	197	removed	271
\@gls@defaulttype: new	74	\DescriptionFootnoteNewAcronymDef:	
\@gls@do@automake@err: new	36	removed	271, 382
\@gls@do@glsunset: new	124	\DescriptionNewAcronymDef:	
\@gls@entry@field: switched to		removed	271, 382
\csuse	169	dua: switched to using	
\@gls@escbsdq: check		\glsuppercase	264, 265
esclocations setting	130	dua-desc: removed \acronymfont	
\@gls@getcounterprefix:		from around long form	265
changed \edef to \def	213	\DUANewAcronymDef: removed	272
\@gls@getgrouptitle: switched		esclocations: default now false	12
to L ^A T _E X3 syntax	244	\findrootlanguage: removed	65
\@gls@glslink@hyper@update@hook:		footnote: switched to using	
new	123	\glsuppercase	266
\@gls@hypergroup: updated to		\FootnoteNewAcronymDef:	
use etoolbox	283	removed	272, 382
\@gls@noist@toolate: new	193	\glossary: removed	39
\@gls@removedoption: new	27	\glossaryentryfield: removed	243
\@gls@requested@languages:		\glossarymark: switched to using	
new	31	\glsuppercase	55
\@gls@restore@glslocal: new	124	\glossarystyle: removed	246
\@gls@sanitizedesc: removed	24	\glossarysubentryfield:	
\@gls@save@glslocal: new	123	removed	243
\@gls@setupsort@clear: new	18	\gls@automake@makegloss: new	36
\@gls@defaultplural: Removed	84	\gls@automake@option: new	36
\@org@Gls@acentryname: saved		\gls@codepage: set default	
original definition of		codepage to utf8 if	
\@Gls@acentryname	171	\inputencodingname not	
\@xp@gls@getcounterprefix:		defined and switched to	
new	213	\ifdefvoid	34
\@xp@gls@hypernumber: new	247	\gls@delimN: renamed \@delimN	
automake: added makegloss and		to \gls@delimN	248
lite options	37	\gls@delimR: renamed \@delimR	
automakegloss: new	38	to \gls@delimR	248
automakeglosslite: new	38	\gls@disablepagerefexpansion:	
compatible-2.07: deprecated	38	added check for esclocations	207
compatible-3.07: deprecated	38	\gls@noidx@nonumberlist: new	81
\compatibleglossentry:		\gls@noidx@glossary: added	
removed	241, 346	\gls@wrglossdisablelocationcmds	
\compatiblesubglossentry:			204
removed	243, 346	\gls@xdy@set@lang: new	33
\CustomAcronymFields:		\gls@addkey: allow expansion in	
removed	273	PDF bookmarks	92

switched to using		
\glssupercase	93
\GlsAddXdyAttribute: remove		
duplicate counterless		
attribute definitions	...	59, 60
\Glsdisp: new	146
\glsdisplay: removed	122
\glsdisplayfirst: removed	...	122
\glsdoparenifnotempty:		
replaced		
\DeclareRobustCommand with		
\newrobustcmd	272
\glsdoshowtarget: new	8
\glsencapwrccontent: new	...	128
\Glsentrydesc: allow expansion		
in PDF bookmarks	172
\Glsentrydescplural: allow		
expansion in PDF		
bookmarks	173
\Glsentryfirst: allow expansion		
in PDF bookmarks	174
\Glsentryfirstplural: allow		
expansion in PDF		
bookmarks	175
\glsentryfmt: obsolete		
commands replaced with		
\glsgenentryfmt	117
\Glsentrylong: allow expansion		
in PDF bookmarks	178
\Glsentrylongpl: allow		
expansion in PDF		
bookmarks	178
\Glsentryname: allow expansion		
in PDF bookmarks	171
\Glsentryplural: allow		
expansion in PDF		
bookmarks	173
\Glsentryprefix: allow		
expansion in PDF		
bookmarks	340
\Glsentryprefixfirst: allow		
expansion in PDF		
bookmarks	339
\Glsentryprefixfirstplural:		
allow expansion in PDF		
bookmarks	340
\Glsentryprefixplural: allow		
expansion in PDF		
bookmarks	340
\Glsentryshort: allow expansion		
in PDF bookmarks	177
\Glsentryshortpl: allow		
expansion in PDF		
bookmarks	178
\Glsentrysymbol: allow		
expansion in PDF		
bookmarks	174
\Glsentrysymbolplural: allow		
expansion in PDF		
bookmarks	174
\Glsentrytext: allow expansion		
in PDF bookmarks	173
\glsentrytitlecase: use		
sentence case for PDF		
bookmark	175
\Glsentryuseri: allow expansion		
in PDF bookmarks	176
\Glsentryuserii: allow		
expansion in PDF		
bookmarks	176
\Glsentryuseriii: allow		
expansion in PDF		
bookmarks	176
\Glsentryuseriv: allow		
expansion in PDF		
bookmarks	177
\Glsentryuserv: allow expansion		
in PDF bookmarks	177
\Glsentryuservi: allow		
expansion in PDF		
bookmarks	177
\glsfielddef: made robust	...	95
\glsfieldedef: made robust	..	94
\glsfieldfetch: made robust	..	95
\glsfieldgdef: made robust	..	95
\glsfieldxdef: made robust	..	94
\glsgenacfmt: switched to using		
\glssupercase	120
\glsgenentryfmt: switched to		
using \glssupercase	118
\glsgetgrouplabel: converted to		
use L ^A T _E X3	245
\glsgetgrouptitle: made		
robust	244
\gls glossarymark: switched to		
\glssupercase	54
\glsifhyper: removed	124
\glsindexingsetting: new	...	15
\glsinlineifhaschildren: new		285

<code>\Glslink</code> : new	125	<code>\SetCustomDisplayStyle</code> :	
<code>\glslocationcstoencap</code> : new	53	removed	273
<code>\glslowercase</code> : new	4	<code>\SetCustomStyle</code> : removed	273
<code>\glsmfuaddmap</code> : new	5	<code>\SetDefaultAcronymStyle</code> :	
<code>\glsmfublocker</code> : new	5	removed	270
<code>\glsmfuexcl</code> : new	5	<code>\SetDescriptionAcronymDisplayStyle</code> :	
<code>\glsnavhyperlink</code> : changed		removed	271
<code>\@glo@type</code> to		<code>\SetDescriptionAcronymStyle</code> :	
<code>\currentglossary</code>	280	removed	271
<code>\glsnavhypertarget</code> : changed		<code>\SetDescriptionDUAAcronymDisplayStyle</code> :	
<code>\@glo@type</code> to		removed	271
<code>\currentglossary</code>	282	<code>\SetDescriptionDUAAcronymStyle</code> :	
<code>\glsnavigation</code> : changed		removed	271
<code>\@glo@type</code> to		<code>\SetDescriptionFootnoteAcronymDisplayStyle</code> :	
<code>\currentglossary</code>	283	removed	271
<code>\glsnoidxprenumberlist</code> : new	82	<code>\SetDescriptionFootnoteAcronymStyle</code> :	
<code>\glsnumberlistloop</code> : corrected		removed	271
parameter number	203	<code>\SetDUADisplayStyle</code> : removed	272
<code>\glsseeitemformat</code> : added test		<code>\SetDUASStyle</code> : removed	272
for short field	215	<code>\SetFootnoteAcronymDisplayStyle</code> :	
<code>\glsentencecase</code> : new	4	removed	271
<code>\glsshowtargetfonttext</code> : new	8	<code>\SetFootnoteAcronymStyle</code> :	
<code>\glsshowtargetinner</code> : new	8	removed	272
<code>\glstexpdfstring</code> : new	170	<code>\SetGenericNewAcronym</code> :	
<code>\glsuppercase</code> : new	4	corrected long plural suffix	256
<code>\GlsWarnAddProtectedPageFmt</code> :		switched to using	
new	208	<code>\glsuppercase</code>	256
<code>\glswrglossdisableanchorcmds</code> :		<code>\setglossentrycompatibility</code> :	
new	208	removed	243
<code>\glswrglossdisablelocationcmds</code> :		<code>\SetSmallAcronymDisplayStyle</code> :	
new	207	removed	272
<code>\glswrglosslocationtarget</code> :		<code>\SetSmallAcronymStyle</code> :	
new	209	removed	272
<code>\glswrglosslocationtextfmt</code> :		<code>\SmallNewAcronymDef</code> :	
new	209	removed	272, 382
<code>\ifglsfieldcseq</code> : made robust	97	<code>sort</code> : added clear	15
<code>\ifglsfielddefeq</code> : made robust	96	4.51 – 2022-11-02	
<code>\ifglsfieldeq</code> : made robust	96	General: switch to	
<code>\ifglsfieldvoid</code> : new	72	<code>\TrackIfKnownLanguage</code>	48
<code>\ifglsresetcurrcount</code> : new	111	<code>\do@wrglossary</code> : added	
<code>\ifglsrallowprimitivemods</code> :		<code>\gls@ifnotmeasuring</code>	206
switch default to false	209	<code>\@gls@@automake@immediate</code> :	
<code>\istfile</code> : removed	204	ensure list is expanded before	
<code>kernelglossredefs</code> : corrected		iterating	197
names of choice variables	40	<code>\@gls@if@measuring</code> : new	106
<code>languages</code> : new	31	<code>alttree</code> : replaced <code>\settowidth</code>	
<code>\makeglossary</code> : removed	39	with <code>\glsmeasurewidth</code>	334, 335
<code>mfirstuc</code> : new	41	<code>\gls@if@measuring</code> : new	106
<code>\SetAcronymStyle</code> : removed	273	<code>\gls@local@stepcounter</code> : new	107
		<code>\gls@org@refstepcounter</code> : new	106

<code>\gls@start@measuring</code> : new ..	107	<code>\Glsentryname</code> : change map to exclusion	171
<code>\gls@stop@measuring</code> : new ...	108	<code>\Glsentryplural</code> : change map to exclusion	173
<code>\glsdohypertarget</code> : added measuring and group	137	<code>\Glsentryshort</code> : change map to exclusion	177
<code>\glsfindwidesttoplevelname</code> : replaced <code>\settowidth</code> with <code>\glsmeasurewidth</code>	333	<code>\Glsentryshortpl</code> : change map to exclusion	178
<code>\glsifmeasuring</code> : new	107	<code>\Glsentrysymbol</code> : change map to exclusion	174
<code>\glsmeasuredepth</code> : new	108	<code>\Glsentrysymbolplural</code> : change map to exclusion	174
<code>\glsmeasureheight</code> : new	108	<code>\Glsentrytext</code> : change map to exclusion	173
<code>\glsmeasurewidth</code> : new	108	<code>\Glsentryuseri</code> : change map to exclusion	176
<code>\glspatchtabularx</code> : switch measuring on and off	108	<code>\Glsentryuserii</code> : change map to exclusion	176
<code>\makeglossaries</code> : ensure list of glossaries is expanded before iterating	199	<code>\Glsentryuseriii</code> : change map to exclusion	176
4.52 – 2022-11-03		<code>\Glsentryuseriv</code> : change map to exclusion	177
<code>\gls@local@stepcounter</code> : corrected spelling of <code>\@ne</code> ..	107	<code>\Glsentryuserv</code> : change map to exclusion	177
4.53 – 2023-09-29		<code>\Glsentryuservi</code> : change map to exclusion	177
<code>\@gls@hypergroup</code> : rewritten in \LaTeX 3	283	<code>\glslabelhypertarget</code> : new ..	138
<code>\glsnavhypergroupdotarget</code> : new	282	<code>\glslabelhypertargetdefs</code> : new	138
<code>\glsnavigationitem</code> : new	283	<code>\glslabelhypertargetprefix</code> : new	138
4.54 – 2024-04-03		<code>\glslabelhypertargetvalue</code> : new	138
General: added check for <code>\l_datatool_measure_hook_tl</code>	107	<code>\glslowercase</code> : ensure argument is expanded	4
<code>\gls@printglossary@begin</code> : new	219	<code>\glsuppercase</code> : ensure argument is expanded	4
<code>\gls@printglossary@end</code> : new ..	220	4.55 – 2024-11-01	
<code>\glsdohyperlinkhook</code> : new ...	139	<code>\@gls@run@output@dir</code> : new ..	42
<code>\glsdohypertargethook</code> : new ..	138	<code>\@gls@run@output@file</code> : new ..	43
<code>\Glsentrydesc</code> : change map to exclusion	172	<code>\@gls@run@unrestricted@shell</code> : new	42
<code>\Glsentrydescplural</code> : change map to exclusion	173	<code>\@gls@runshell</code> : new	42
<code>\Glsentryfirst</code> : change map to exclusion	174	<code>\gls@set@xr@key</code> : added check for empty value	80
<code>\Glsentryfirstplural</code> : change map to exclusion	175	locales: new	31
<code>\Glsentryfull</code> : change map to exclusion	178	4.56 – 2025-03-03	
<code>\Glsentrylong</code> : change map to exclusion	178	General: store original sort as well as label in actual part	221
<code>\Glsentrylongpl</code> : change map to exclusion	178		

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the definition; numbers in *roman* refer to the pages where the entry is used.

Symbols	
<code>\!</code>	131, 134, 192, 459, 460, 708, 710, 711, 741, 962, 964, 995, 1117, 1119, 1151
<code>\"</code>	26, 131–134, 136, 396, 457–459, 461, 646, 708–711, 713, 900, 961–964, 966, 967, 1055, 1116–1119, 1121
<code>\#</code>	136, 186, 484, 485, 712, 713, 736, 966, 991, 1121, 1146, 1147
<code>\%</code>	183, 189, 482, 486, 587–589, 734, 738, 840–842, 988, 993, 1144, 1148
<code>\&</code>	48, 180, 407, 481, 657, 733, 913, 987, 1067, 1143
<code>\'</code>	26, 396, 646, 900, 1055
<code>\.</code>	13, 26, 389, 396, 639, 646, 893, 900, 1048, 1055
<code>\=</code>	26, 396, 646, 900, 1055
<code>\?</code>	131–133, 191, 457–459, 488, 708–710, 740, 961, 963, 995, 1116, 1118, 1150
<code>\@</code>	88, 429, 680, 936, 1090
<code>\@@delimN</code>	519, 771
<code>\@do@wrglossary</code>	200, 209, 212, <u>212</u> , 494, 500, 501, 746, 751, 752, 1001, 1007, 1008, 1157, 1163, 1164
<code>\@do@esc@wrglossary</code>	208, <u>209</u> , 499, 500, 751, 752, 1007, 1162, 1163
<code>\@do@noesc@wrglossary</code>	208, <u>209</u> , 499, 751, 1007, 1162
<code>\@do@wrglossary</code>	181, 206, <u>208</u> , 481, 498, 499, 733, 750, 751, 988, 1005, 1007, 1143, 1161, 1162
<code>\@do@write@glslabels</code>	35, <u>35</u> , 402, 652, 907, 1061
<code>\@glo@sortentries</code>	226, 229, 230
<code>\@glo@assign@sortkey</code>	201, <u>238</u> , 495, 513, 747, 765, 1002, 1021, 1158, 1176
<code>\@glo@list</code>	67, 418, 668, 924, 1079
<code>\@glo@sort</code>	26, 396, 646, 900, 1055
<code>\@glo@type</code>	217, 503, 755, 1011, 1166, 1167
<code>\@glossarysec</code>	10, <u>10</u> , 56, 388, 411, 637, 638, 661, 892, 917, 1046, 1071, 1072
<code>\@glossaryseclabel</code>	10, <u>10</u> , 11, 56, 237, 388, 411, 513, 638, 661, 764, 765, 892, 917, 1020, 1046, 1047, 1071, 1072, 1176
<code>\@glossarysecstar</code>	10, <u>10</u> , 11, 56, 237, 388, 411, 513, 638, 661, 764, 765, 892, 917, 1020, 1046, 1047, 1071, 1176
<code>\@gls@default@acr@desclplural</code>	<u>270</u> , 785
<code>\@gls@default@acr@firstplural</code>	<u>270</u> , 785
<code>\@gls@default@acr@plural</code>	<u>270</u> , 785
<code>\@gls@checkactual</code>	135, 461, 712, 965, 966, 1120, 1121
<code>\@gls@checkbar</code>	134, 460, 711, 965, 1119, 1120
<code>\@gls@checkescactual</code>	133, 458, 459, 709, 710, 963, 1118
<code>\@gls@checkescbar</code>	133, 459, 710, 964, 1118, 1119
<code>\@gls@checkesclevel</code>	134, 459, 460, 710, 711, 964, 1119
<code>\@gls@checkescquote</code>	132, 192, 193, 458, 489, 490, 709, 741, 742, 963, 996, 1117, 1118, 1151, 1152
<code>\@gls@checklevel</code>	135, 460, 461, 711, 712, 965, 1120
<code>\@gls@checkquote</code>	132, 190, 191, 457, 458, 488, 708, 709, 739, 740, 962, 994, 1117, 1150
<code>\@gls@codepage</code>	195, 196, 997–999, 1153, 1154
<code>\@gls@default@entryfmt</code>	<u>358</u> , 446, 452, 453, 557, 697, 703, 810
<code>\@gls@expand@field</code>	24, 85, <u>85</u> , 91, 92, 395, 427, 428, 431, 533– 536, 538, 540, 542, 544, 578–

582, 645, 678, 681, 682, 785, 787, 788, 790, 792, 793, 795, 797, 832–835, 899, 934, 937, 938, 1054, 1088, 1089, 1092, 1093	\@ACRlong 166, 253, 366, 475, 521, 565, 726, 773, 818, 980, 1027, 1136, 1183
\@gls@extramakeindexopts 190, 198, 487, 493, 739, 745, 994, 1000, 1149, 1156	\@ACRlongpl 168, 169, 253, 476, 522, 727, 774, 981, 1027, 1137, 1183
\@gls@fixbraces 214, 214, 502, 754, 1009, 1010, 1165	\@ACRshort 162, 163, 253, 365, 472, 521, 564, 724, 773, 818, 978, 1027, 1134, 1183
\@gls@navhypertarget 282, 282, 1042	\@ACRshortpl 164, 253, 473, 474, 522, 725, 774, 979, 1027, 1135, 1183
\@gls@noexpand@assign@default@plurals 270, 785, 832	\@Acrfull . . 253, 521, 773, 1028, 1183
\@gls@noexpand@field 24, 84, 85, 395, 427, 645, 677, 678, 899, 933, 934, 1054, 1088	\@Acrfullpl . 254, 521, 773, 1028, 1184
\@gls@noidx@no@sanitizesort 26, 396, 646, 900, 1055	\@Acrlong 166, 252, 365, 474, 521, 565, 726, 773, 818, 980, 1027, 1135, 1136, 1183
\@gls@noidx@nosanitizesort 26, 203, 396, 496, 646, 748, 900, 1003, 1055, 1159	\@Acrlongpl . . . 168, 253, 475, 521, 727, 773, 981, 1027, 1137, 1183
\@gls@noidx@sanitizesort 25, 396, 645, 900, 1054	\@Acrshort 162, 364, 472, 564, 724, 817, 978, 1133
\@gls@nosanitizesort 25, 25, 203, 395, 396, 496, 645, 748, 899, 900, 1003, 1054, 1159	\@Acrshortpl . 164, 473, 725, 979, 1134
\@gls@sanitizesort 25, 25, 203, 395, 396, 496, 645, 748, 899, 900, 1003, 1054, 1159	\@Alph 207, 211, 499, 500, 751, 752, 1006–1008, 1162, 1163
\@gls@xdycheckbackslash 137, 462, 713, 714, 967, 1122	\@GLS . . . 142, 464, 715, 716, 969, 1125
\@gls@xdycheckquote 136, 137, 461, 713, 966, 967, 1121, 1122	\@GLS@ 142, 142, 344, 464, 604, 716, 857, 969, 1125
\@gls@localreset 110, 110, 112, 442, 444, 693, 694, 950, 951, 1105, 1106	\@GLSdesc . . . 152, 468, 720, 973, 1129
\@gls@localunset 109, 109, 112, 442, 443, 693, 694, 950, 951, 1105, 1106	\@GLSdesc@ 152, 368, 468, 567, 720, 820, 973, 1129
\@gls@reset . . . 110, 110, 112, 442, 443, 693, 694, 950, 951, 1105, 1106	\@GLSdescplural 153, 469, 720, 974, 1129
\@gls@showtarget 7, 8, 9, 386, 387, 636, 637, 890, 891, 1045	\@GLSdescplural@ 153, 368, 469, 567, 720, 820, 974, 1129
\@gls@unset . . . 109, 110, 112, 442, 443, 693, 694, 950, 951, 1105, 1106	\@GLSfirst . . 148, 467, 718, 972, 1127
\@newglossaryentry@defcounters 111, 111, 443, 694, 951, 1105, 1106	\@GLSfirst@ 148, 367, 467, 566, 718, 819, 972, 1127
\@this@glo@ . . 67, 418, 668, 924, 1079	\@GLSfirstplural 150, 467, 719, 973, 1128
\@ACRfull 254, 521, 773, 1028, 1183, 1184	\@GLSfirstplural@ 150, 367, 468, 566, 719, 820, 973, 1128
\@ACRfullpl 255, 521, 522, 773, 1028, 1184	\@GLSname . . . 151, 468, 719, 973, 1129
	\@GLSname@ 151, 368, 468, 567, 720, 820, 973, 1129
	\@GLSpl . 144, 144, 465, 717, 970, 1126
	\@GLSpl@ 144, 345, 465, 605, 717, 858, 970, 1126
	\@GLSplural . . 149, 467, 719, 972, 1128
	\@GLSplural@ 149, 367, 467, 566, 719, 819, 972, 1128
	\@GLSsymbol . . 154, 469, 721, 974, 1130

<code>\@GLSsymbol@</code>	154,	<code>\@Glsdescplural</code>	
	369 , 469 , 567 , 721 , 820 , 974 , 1130		. . . 152, 153, 468, 720, 974, 1129
<code>\@GLSsymbolplural</code>		<code>\@Glsdescplural@</code>	153,
 155, 469, 721, 975, 1130		368 , 468 , 567 , 720 , 820 , 974 , 1129
<code>\@GLSsymbolplural@</code>	155,	<code>\@Glsdisp</code>	146, 146 , 971, 1126
	369 , 469 , 567 , 721 , 821 , 975 , 1130	<code>\@Glsentryfield</code>	
<code>\@GLStext</code> 146, 147, 466, 718, 971, 1127			. 92, 170 , 172–178, 339, 340,
<code>\@GLStext@</code>	147,		938, 982–986, 1093, 1138–1142
	366 , 466 , 566 , 718 , 819 , 971 , 1127	<code>\@Glsfirst</code> 147, 148, 467, 718, 972, 1127	
<code>\@GLSuseri</code> . . 156, 470, 721, 975, 1131		<code>\@Glsfirst@</code>	148,
<code>\@GLSuseri@</code>	156,		367 , 467 , 566 , 718 , 819 , 972 , 1127
	369 , 470 , 568 , 721 , 821 , 975 , 1131	<code>\@Glsfirstplural</code>	
<code>\@GLSuserii</code> . . 157, 470, 722, 975, 1131			. . . 149, 150, 467, 719, 972, 1128
<code>\@GLSuserii@</code>	157,	<code>\@Glsfirstplural@</code>	
	370 , 470 , 568 , 722 , 821 , 975 , 1131		. . . 150, 367 , 467 , 719 , 972 , 1128
<code>\@GLSuseriii</code> . 158, 470, 722, 976, 1131		<code>\@Glsname</code> 150, 151, 468, 719, 973, 1128	
<code>\@GLSuseriii@</code>	158,	<code>\@Glsname@</code>	151,
	370 , 470 , 568 , 722 , 821 , 976 , 1131		368 , 468 , 566 , 719 , 820 , 973 , 1128
<code>\@GLSuseriv</code> . . 159, 471, 722, 976, 1132		<code>\@Glspl</code>	143, 465, 716, 970, 1125
<code>\@GLSuseriv@</code>	159,	<code>\@Glspl@</code>	113, 116,
	370 , 471 , 568 , 722 , 821 , 976 , 1132		143, 144 , 344, 444, 446, 465,
<code>\@GLSuserv</code> . . 160, 471, 723, 977, 1132			603, 604, 695, 696, 716, 857,
<code>\@GLSuserv@</code>	160,		952, 954, 970, 1107, 1108, 1125
	371 , 471 , 569 , 723 , 822 , 977 , 1132	<code>\@Glsplural</code>	
<code>\@GLSuservi</code> . . 161, 471, 723, 977, 1133			. . . 148, 149, 467, 719, 972, 1128
<code>\@GLSuservi@</code>	161,	<code>\@Glsplural@</code>	149,
	371 , 471 , 569 , 723 , 822 , 977 , 1133		367 , 467 , 566 , 719 , 819 , 972 , 1128
<code>\@Gls</code>	141, 463, 715, 969, 1124	<code>\@Glsymbol</code>	
<code>\@Gls@</code>	113,		. . . 153, 154, 469, 720, 974, 1130
	115, 141, 141 , 343, 444, 445,	<code>\@Glsymbol@</code>	154,
	463, 603, 695, 696, 715, 856,		369 , 469 , 567 , 720 , 820 , 974 , 1130
	952, 953, 969, 1107, 1108, 1124	<code>\@Glsymbolplural</code>	
<code>\@Gls@link</code>	125, 126, 958, 1113		. . . 154, 155, 469, 721, 974, 1130
<code>\@Gls@acentryname</code>		<code>\@Glsymbolplural@</code>	155,
 171 , 256, 477, 522,		369 , 469 , 567 , 721 , 821 , 974 , 1130
	728, 774, 983, 1029, 1138, 1184	<code>\@Glstext</code> . . . 147, 466, 718, 971, 1127	
<code>\@Gls@entry@field</code>	170,	<code>\@Glstext@</code>	147,
	170 , 171, 172, 432, 476–480,		366 , 466 , 566 , 718 , 819 , 971 , 1127
	682, 728–731, 982, 983, 1138, 1139	<code>\@Glsuser@i</code> 369–371, 568, 569, 821, 822	
<code>\@Gls@entryname</code>		<code>\@Glsuseri</code> 155, 156, 470, 721, 975, 1130	
	171, 171 , 256, 477, 522, 728,	<code>\@Glsuseri@</code>	
	774, 982, 983, 1029, 1138, 1184		. . . 156, 369 , 470 , 721 , 975 , 1130
<code>\@GlsSetXdyFirstLetterAfterDigits</code>		<code>\@Glsuserii</code>	
	. . . 183, 482, 734, 988, 989, 1144		. . . 156, 157, 470, 722, 975, 1131
<code>\@GlsSetXdyNumberGroupOrder</code> 183,		<code>\@Glsuserii@</code>	
	184, 482, 483, 734, 989, 1144, 1145		. . . 157, 370 , 470 , 722 , 975 , 1131
<code>\@Glsdesc</code> 151, 152, 468, 720, 973, 1129		<code>\@Glsuseriii</code>	
<code>\@Glsdesc@</code>	152,		. . . 157, 158, 470, 722, 976, 1131
	368 , 468 , 567 , 720 , 820 , 973 , 1129	<code>\@Glsuseriii@</code>	
			. . . 158, 370 , 470 , 722 , 976 , 1131

<code>\@Glsuseriv</code>	1013, 1042, 1082, 1083, 1107, 1123, 1156, 1157, 1160, 1166, 1168
... 158, 159, 471, 722, 976, 1132	
<code>\@Glsuseriv@</code>	<code>\@backslashchar</code>
... 159, <u>370</u> , 471, 722, 976, 1132	130, 131, 137, 192, 456, 462, 707, 708, 713, 714, 741, 961, 962, 967, 995, 1116, 1117, 1122, 1151
<code>\@Glsuserv</code> 159, 160, 471, 723, 976, 1132	<code>\@bsphack</code> .. 206, 498, 750, 1005, 1161
<code>\@Glsuserv@</code>	<code>\@cGls</code> .. 115, <u>115</u> , 445, 696, 953, 1108
... 160, <u>371</u> , 471, 723, 976, 1132	<code>\@cGls@</code>
<code>\@Glsuservi</code>	113, 115, <u>115</u> , 444, 445, 695, 696, 952, 953, 1107, 1108
... 160, 161, 471, 723, 977, 1132	<code>\@cGlspl</code> 116, 445, 696, 953, 1108
<code>\@Glsuservi@</code>	<code>\@cGlspl@</code> 113, 116, <u>116</u> , 444– 446, 695, 696, 952, 953, 1107, 1108
... 161, <u>371</u> , 471, 723, 977, 1132	<code>\@cclv</code>
<code>\@Mi</code>	301, 614, 615, 868
301, 614, 868	<code>\@cgls</code> .. 114, <u>114</u> , 445, 696, 953, 1107
<code>\@PGLS</code>	<code>\@cgls@</code>
344, <u>344</u> , 604, 857	112, 114, <u>114</u> , 444, 445, 694, 696, 951, 953, 1106, 1108
<code>\@PGLS@</code>	<code>\@cglspl</code>
344, <u>344</u> , 604, 857	115, <u>115</u> , <u>116</u> , 445, 696, 953, 1108
<code>\@PGLSpl</code>	<code>\@cglspl@</code> 113, 115, <u>115</u> , 444, 445, 695, 696, 952, 953, 1106, 1108
345, <u>345</u> , 604, 857	<code>\@chapter</code> 46, 406, 656, 912, 1067
<code>\@PglS</code>	<code>\@classoptionslist</code>
342, <u>342</u> , 603, 856	42, 405, 655, 911, 1065
<code>\@PglS@</code>	<code>\@closegls</code>
342, <u>342</u> , 603, 856	194, <u>194</u> , 195, 490, 491, 742, 743, 997, 998, 1153
<code>\@PglSpl</code>	<code>\@colht</code>
343, <u>343</u> , 603, 856	301, 615, 868
<code>\@PglSpl@</code>	<code>\@colroom</code>
343, <u>343</u> , 603, 856	301, 615, 868
<code>\@Roman</code>	<code>\@currentlabelname</code>
207, 211, 499, 500, 751, 752, 1006, 1007, 1162, 1163	11, 237, 388, 513, 638, 765, 892, 1020, 1047, 1176
<code>\@acrfull</code> 253, <u>253</u> , 520, 772, 1027, 1183	<code>\@curroptions</code> . 42, 405, 655, 911, 1065
<code>\@acrfullpl</code> . 254, 521, 773, 1028, 1184	<code>\@declaredoptions</code>
<code>\@acrlong</code>	42, 405, 655, 911, 1065
165, 252, <u>365</u> , 474, 520, 565, 725, 726, 772, 818, 979, 980, 1027, 1135, 1183	<code>\@delimN</code>
<code>\@acrlongpl</code> ... 167, 253, 475, 521, 727, 773, 981, 1027, 1136, 1183	519, 771
<code>\@acrshort</code>	<code>\@delimR</code>
161, 252, <u>364</u> , 472, 520, 521, 564, 723, 772, 773, 817, 977, 1027, 1133, 1183	519, 771
<code>\@acrshortpl</code> .. 163, 253, 473, 521, 724, 773, 978, 1027, 1134, 1183	<code>\@disable@onlypremakeg</code>
<code>\@addtoacronymlists</code>	46, 198, 406, 493, 657, 745, 912, 1000, 1067, 1156
21, 393, 643, 897, 1052	<code>\@disable@premakecs</code>
<code>\@addtoacronymlists</code>	46, <u>47</u> , 407, 657, 912, 1067
<u>21</u>	<code>\@disabled@glsaddxdycounters</code> <u>59</u> , 60, 413, 663, 664, 919, 1073, 1074
<code>\@afterheading</code> 289, 591, 608, 844, 862	<code>\@do@addcounter</code>
<code>\@alph</code>	58, 412, 413, 663, 918, 1073
207, 211, 499, 500, 751, 752, 1006–1008, 1162, 1163	<code>\@do@auxoutstuff</code>
<code>\@arabic</code>	220, 221, 505, 757, 1012, 1013, 1168
207, 211, 499, 500, 751, 752, 1006, 1007, 1162, 1163	<code>\@do@glossentry</code> .. 515, 549, 767, 802
<code>\@auxout</code>	<code>\@do@gls@checkesc</code> 131, 708, 962, 1117
74, 76, 114, 138, 198, 200, 201, 204, 216, 220, 282, 422, 445, 493–495, 497, 503, 505, 605, 672, 673, 695, 696, 745–747, 749, 755, 757, 858, 928, 929, 952, 953, 1000, 1002, 1004, 1010, 1012,	

<code>\@do@gls@getcounterprefix</code>	1052, 1063, 1065–1067, 1073,
. 213, 499,	1074, 1076, 1078, 1079, 1098,
500, 751, 752, 1008, 1009, 1164	1102, 1114, 1115, 1124–1126,
<code>\@do@gls@islistofacronyms</code>	1133–1137, 1145, 1147, 1149,
. 21, 393, 643, 897, 1052	1153–1155, 1160, 1161, 1164, 1180
<code>\@do@glssee</code> 104, 439, 690, 946, 1101	<code>\@end@fixbraces</code>
<code>\@do@ifinlist</code> 58, 412, 662, 918, 1073 214, 502, 754, 1009, 1010, 1165
<code>\@do@newglossaryentry</code>	<code>\@endfortrue</code>
. 256, 522, 532–545, 578–582, 31, 70, 90, 400, 420, 430,
774, 784–799, 831–835, 1029, 1185	605, 650, 670, 681, 858, 904,
<code>\@do@seeglossary</code> 200,	925, 937, 1042, 1059, 1080, 1092
214, 214, 494, 502, 746, 753,	<code>\@esphack</code> 206, 498, 750, 1005, 1161
754, 1002, 1009, 1010, 1157, 1165	<code>\@expandtwoargs</code> 42, 405, 655, 911, 1065
<code>\@do@subglossentry</code> 516, 549, 768, 802	<code>\@firstofone</code> 26,
<code>\@do@wrglossary</code>	35, 287, 396, 397, 402, 646,
127, 206, 455, 498, 585, 705,	652, 861, 900, 901, 906, 1055, 1061
750, 838, 959, 1005, 1114, 1161	<code>\@firstofthree</code> 124, 140, 143,
<code>\@do@write@glslabels</code> 35,	145, 161, 163, 165, 167, 364–
36, 402, 652, 906, 907, 1061, 1062	366, 453, 463, 464, 466, 472–
<code>\@do@writeaux@info</code>	475, 564, 565, 704, 715–717,
. 216, 503, 755, 1010, 1166	723, 725–727, 817–819, 958,
<code>\@doendpe</code> 220, 1168	968, 970, 971, 977, 978, 980,
<code>\@domakeglossaries</code> 34,	981, 1112, 1124–1126, 1133–1136
34, 35, 197, 200, 401, 402, 493,	<code>\@firstoftwo</code> 29–31, 89, 90, 107, 124,
494, 651, 652, 745, 746, 906,	143, 144, 163–165, 167–169,
1000, 1001, 1060, 1061, 1155, 1157	399, 400, 429, 430, 453, 464,
<code>\@ehc</code> 301, 614, 868	465, 473–476, 648–650, 680,
<code>\@empty</code> 14, 18, 21, 38, 41, 42,	681, 704, 716, 717, 724, 725,
46, 58, 59, 63, 66, 67, 100, 105,	727, 728, 903, 904, 936, 937,
106, 128, 129, 140–144, 161–	948, 958, 970, 978, 979, 981,
169, 184, 187, 189, 195, 196,	982, 1058, 1059, 1091, 1092,
204, 206, 213, 245, 364–366,	1103, 1112, 1125, 1126, 1134–1137
390, 392, 393, 403, 405, 406,	<code>\@for</code> 31, 42, 46, 48, 58, 60, 67,
412, 413, 415, 417, 418, 437,	89, 90, 130, 184–186, 197–199,
440, 455, 463–465, 472–476,	207, 215, 226, 227, 257, 400,
483, 485, 487, 491, 492, 497,	405, 406, 412, 413, 418, 429,
498, 501, 517, 533, 535–542,	430, 456, 483, 484, 493, 494,
544, 546, 564, 565, 585, 587–	499, 502, 506, 523, 533, 535,
589, 639, 641, 643, 653, 655,	537, 538, 540, 543, 544, 546,
656, 662, 663, 666, 668, 687,	585, 605, 606, 650, 655, 657,
688, 691, 706, 715–717, 723–	663, 668, 680, 681, 707, 735,
728, 735, 737, 739, 742–744,	736, 745, 746, 751, 754, 757,
749, 750, 753, 769, 785, 787,	758, 775, 785, 788, 789, 791,
789, 791, 793–795, 797, 799,	793, 796, 797, 799, 839, 858,
817–819, 839, 841, 842, 893,	859, 904, 911–913, 918, 919,
895–897, 908–912, 918, 919,	924, 936, 937, 961, 989–991,
921, 923, 924, 943, 944, 947,	999–1001, 1006, 1010, 1013,
959, 960, 968–970, 977–982,	1030, 1042, 1059, 1065, 1067,
989, 992, 993, 997–999, 1004,	1068, 1073, 1074, 1078, 1079,
1005, 1009, 1024, 1048, 1050,	1091, 1092, 1116, 1145, 1146,

1155, 1156, 1161, 1165, 1169, 1186
 \@glo@desc .. 102, 438, 689, 945, 1100
 \@glo@symbol 103, 439, 690, 946, 1100
 \@glo@access .. 347, 350, 355, 549,
 551, 552, 554, 555, 803–805, 808
 \@glo@addchildren ..
 227, 227, 231, 506, 509,
 758, 761, 1013, 1016, 1169, 1172
 \@glo@assign@sortkey 199, 201, 238,
 494, 495, 513, 745, 747, 765,
 1001, 1002, 1021, 1156, 1158, 1176
 \@glo@autosee ..
 ... 103, 103, 439, 690, 946, 1101
 \@glo@autoseehook ..
 ... 104, 104, 439, 690, 946, 1101
 \@glo@check@mkidxrangear ..
 129, 212, 456, 501, 585, 707,
 752, 838, 960, 1008, 1115, 1164
 \@glo@check@sortallowed ..
 16, 16, 17–19,
 199, 203, 390–392, 494, 496,
 640–642, 746, 748, 894–896,
 1001, 1003, 1049–1051, 1156, 1159
 \@glo@childlist ..
 227, 506, 758, 1013, 1169
 \@glo@counter .. 80, 99, 102,
 425, 436, 438, 675, 687, 689,
 931, 943, 945, 1085, 1097, 1100
 \@glo@counterprefix 204, 209, 212,
 213, 245, 248, 497, 499–501,
 517, 519, 749, 751–753, 769,
 771, 1004, 1007–1009, 1024,
 1026, 1160, 1162–1164, 1180, 1181
 \@glo@default@sorttype ..
 15, 15, 201,
 229, 230, 390, 495, 508, 509,
 640, 747, 759–761, 894, 1002,
 1015, 1016, 1049, 1158, 1171, 1172
 \@glo@defaultcounter ..
 102, 438, 689, 945, 1100
 \@glo@desc .. 79,
 97, 98, 100, 102, 424, 435–439,
 674, 686, 687, 689, 930, 942,
 943, 945, 1085, 1097, 1098, 1100
 \@glo@descaccess ..
 348, 350, 351, 550–552, 803–805
 \@glo@descplural ..
 ... 79, 97, 98, 424, 435, 436,
 674, 686, 930, 942, 943, 1085, 1097
 \@glo@descpluralaccess ..
 348, 350, 351, 550–552, 803–805
 \@glo@do@sortentries 226, 227, 506,
 757, 758, 1013, 1014, 1169, 1170
 \@glo@entry ..
 ... 181, 182, 482, 733, 988, 1144
 \@glo@entryprefix ..
 338, 339, 600, 601, 853, 854
 \@glo@entryprefixfirst ..
 338, 339, 600, 601, 853, 854
 \@glo@entryprefixfirstplural ..
 338, 339, 600, 601, 853, 854
 \@glo@entryprefixplural ..
 338, 339, 600, 601, 853, 854
 \@glo@esclabel .. 105, 106,
 440, 441, 691, 692, 947, 948, 1102
 \@glo@etext .. 446–448, 697–699
 \@glo@first .. 79, 98, 101, 102,
 424, 436, 438, 439, 542, 581,
 674, 687, 689, 795, 835, 930,
 943, 945, 1085, 1097, 1099, 1100
 \@glo@firstaccess ..
 347, 350, 550–552, 803–805
 \@glo@firstplural ..
 .. 79, 98, 101, 102, 424, 436,
 438, 581, 674, 687, 689, 835,
 930, 943, 945, 1085, 1097, 1099
 \@glo@firstpluralaccess ..
 347, 350, 550–552, 803–805
 \@glo@grabfirst .. 232, 232, 510, 762, 1017, 1173
 \@glo@label .. 81, 83, 91,
 92, 94–104, 111, 179, 180, 333,
 339, 350, 351, 426, 431, 433–
 439, 443, 480, 481, 551–553,
 601, 632, 633, 676, 677, 682,
 684–690, 694, 732, 733, 805,
 806, 854, 886, 931–933, 938,
 940–946, 951, 987, 988, 1086,
 1087, 1092–1101, 1106, 1142, 1143
 \@glo@list .. 104, 439, 690, 946, 1101
 \@glo@long .. 71,
 83, 99, 102, 420, 426, 436, 438,
 670, 676, 687, 689, 926, 932,
 943, 945, 1081, 1087, 1098, 1100
 \@glo@longaccess ..
 348, 350, 351, 550–552, 803–805
 \@glo@longpl .. 83, 99,
 102, 426, 436, 438, 532, 534,
 536, 538, 539, 542, 543, 578,

676, 687, 689, 784, 785, 787, 788, 790, 792, 795, 796, 831, 932, 943, 945, 1087, 1098, 1100	578–581, 803–805, 831–833, 835
\@glo@longpluralaccess	\@glo@shortpl 83, 99, 102,
348, 350, 351, 550–552, 803–805	426, 436, 438, 532, 534, 536,
\@glo@name . . . 16, 78, 84, 98, 101, 102, 391, 424, 427, 436–438, 640, 674, 677, 686, 688, 689, 894, 930, 933, 942, 944, 945, 1049, 1085, 1088, 1097, 1099, 1100	538, 539, 542, 543, 578, 581, 676, 687, 689, 784–786, 788, 790, 792, 795, 796, 831, 835, 932, 943, 945, 1087, 1098, 1100
\@glo@no@assign@sortkey	\@glo@shortpluralaccess
. 199, <u>238</u> , 494, 513, 745, 765, 1001, 1021, 1156, 1176	348, 350, 351, 550–552, 803–805
\@glo@nonumberlist	\@glo@sort . 16, 18, 25, 26, 79, 99,
. 82, 425, 426, 676, 932, 1086	101, 105, 106, 391, 392, 396, 424, 436, 438, 440, 441, 640, 642, 645, 646, 674, 687, 688, 691, 692, 894–896, 900, 930, 943, 944, 947, 948, 1049, 1051, 1054, 1055, 1085, 1097, 1099, 1102
\@glo@numfmt	\@glo@sortedinsert 228, <u>228</u> , 506, 507, 758, 759, 1014, 1169, 1170
213, 501, 585, 753, 838, 1008, 1164	\@glo@sortentries 226, <u>226</u> ,
\@glo@parent	505, 508, 509, 757, 759, 760, 1013, 1015, 1016, 1168, 1171, 1172
. . . 18, 81, 99–101, 105, 106, 227, 228, 392, 425, 436, 437, 440, 441, 506, 641, 675, 687, 688, 691, 692, 758, 895, 931, 943, 944, 947, 948, 1014, 1050, 1086, 1097–1099, 1102, 1169, 1170	\@glo@sorthandler@case
\@glo@plural 79, 98, 101, 424, 436– 438, 580, 674, 687–689, 833, 930, 943–945, 1085, 1097, 1099 226, <u>229</u> , 507, 508, 759, 760, 1014, 1016, 1170, 1171
\@glo@pluralaccess	\@glo@sorthandler@letter
. 347, 350, 550–552, 803–805 226, <u>228</u> , 507, 508, 759, 760, 1014, 1015, 1170, 1171
\@glo@prefix 12, 81, 99, 105, 106, 129, 130, 212, 389, 425, 436, 440, 441, 456, 501, 585, 638, 675, 687, 691, 692, 707, 753, 838, 893, 931, 943, 947, 948, 960, 961, 1008, 1047, 1086, 1097, 1102, 1115, 1164	\@glo@sorthandler@nocase
\@glo@range 226, <u>229</u> , 507, 509, 759, 760, 1015, 1016, 1170, 1172
212, 501, 585, 753, 838, 1008, 1164	\@glo@sorthandler@word
\@glo@see 80, 99, 103, 104, 425, 436, 439, 675, 687, 690, 931, 943, 946, 1085, 1097, 1101 226, <u>228</u> , 507, 508, 759, 1014, 1015, 1170, 1171
\@glo@seeautonumberlist	\@glo@sortinghandler
. 12, <u>12</u> , 81, 389, 425, 638, 675, 892, 931, 1047, 1086 226, 228, 505, 507, 757, 759, 1013, 1014, 1168, 1170
\@glo@short . . . 71, 72, 83, 99, 102, 420, 426, 436, 438, 581, 671, 676, 687, 689, 835, 926, 932, 943, 945, 1081, 1087, 1098, 1100	\@glo@sortinglist 226, 228, 231, 505–507, 509, 757, 758, 761, 1013, 1014, 1016, 1168–1170, 1172
\@glo@shortaccess	\@glo@sortmacro@case
. 348, 350, 351, 550–552, <u>230</u> , 508, 760, 1016, 1171

<code>\@glo@sortmacro@standard</code>	205, 218–221, 226, 228, 231,
. 230 , 508, 760, 1015, 1171	232, 236, 237, 256, 388, 392,
<code>\@glo@sortmacro@use</code>	424, 436–439, 481, 482, 493,
. 231 , 509, 761, 1017, 1172	497, 498, 504–506, 509, 510,
<code>\@glo@sortmacro@word</code>	512, 513, 522, 533, 535, 536,
. 229 , 507, 759, 1015, 1171	538, 540, 542, 544, 546, 605,
<code>\@glo@sorttype</code> 201,	606, 638, 641, 674, 686–690,
231–233, 238, 495, 509, 510,	733, 745, 749, 750, 755–758,
513, 747, 761, 762, 765, 1002,	761, 762, 764, 765, 774, 785,
1017, 1021, 1158, 1172, 1173, 1176	787, 789, 791, 793, 795, 797,
<code>\@glo@storeentry</code> . . . 16–18, 105 ,	799, 858, 859, 892, 895, 896,
390–392, 440, 640, 641, 691,	930, 943–946, 988, 1000, 1004,
894–896, 947, 1049, 1050, 1102	1005, 1011–1014, 1016, 1017,
<code>\@glo@suffix</code> 129, 130,	1019, 1020, 1029, 1047, 1050,
212, 456, 501, 585, 707, 753,	1085, 1097, 1098, 1100, 1101,
838, 960, 961, 1008, 1115, 1164	1143, 1144, 1156, 1160, 1167–
<code>\@glo@symbol</code>	1169, 1172, 1173, 1175, 1176, 1184
. . . 71, 80, 98, 103, 420, 424,	<code>\@glo@types</code> . . . 35, 67, 74 , 76, 110,
436, 439, 537, 541, 670, 674,	111, 181, 182, 197–199, 278,
687, 690, 790, 794, 926, 930,	333, 402, 418, 422, 442, 443,
943, 946, 1081, 1085, 1097, 1100	482, 493, 494, 548, 632, 652,
<code>\@glo@symbolaccess</code> 347,	668, 672, 693, 694, 733, 745,
350, 550–552, 581, 803–805, 834	746, 801, 886, 907, 924, 927,
<code>\@glo@symbolplural</code> . . 80, 98, 103,	928, 950, 951, 988, 999–1001,
424, 436, 439, 674, 687, 690,	1041, 1061, 1078, 1079, 1082,
930, 943, 946, 1085, 1097, 1100	1083, 1105, 1144, 1155–1157, 1197
<code>\@glo@symbolpluralaccess</code>	<code>\@glo@useri</code> 82, 99, 102,
347, 350, 351, 550–552, 803–805	426, 436, 438, 676, 687, 689,
<code>\@glo@text</code> 79, 98, 101, 103,	932, 943, 945, 1087, 1097, 1100
141–145, 170–172, 424, 436,	<code>\@glo@useriaccess</code>
438, 439, 463–466, 476, 477,	348, 350, 351, 550–552, 803–805
538, 580, 601, 674, 687, 688,	<code>\@glo@userii</code> 82, 99, 102,
690, 715–717, 728, 729, 790,	426, 436, 438, 676, 687, 689,
833, 854, 930, 943, 944, 946,	932, 943, 945, 1087, 1098, 1100
969–971, 982, 983, 1085, 1097,	<code>\@glo@useriiaaccess</code> 348,
1099, 1100, 1124–1126, 1138, 1139	350, 351, 550–552, 803, 804, 806
<code>\@glo@textaccess</code> 347, 350, 549, 551,	<code>\@glo@useriii</code> 82, 99, 102,
552, 578–580, 803–805, 831–834	426, 436, 438, 676, 687, 689,
<code>\@glo@thislabel</code>	932, 943, 945, 1087, 1098, 1100
. . . 104, 440, 690, 691, 947, 1101	<code>\@glo@useriiiaccess</code> 348,
<code>\@glo@thislettergrp</code>	350, 351, 550–552, 803, 804, 806
. 225, 232–234, 510, 511,	<code>\@glo@useriv</code> 83, 99, 102,
762, 763, 1017, 1018, 1173, 1174	426, 436, 438, 676, 687, 689,
<code>\@glo@thisvalue</code>	932, 943, 945, 1087, 1098, 1100
72, 73, 421, 671, 927, 1081, 1082	<code>\@glo@userivaccess</code>
<code>\@glo@tmp</code> 74, 91, 92, 349–351, 550–552, 804–806
213, 431, 501, 682, 753, 928,	<code>\@glo@useriv</code> 83, 99, 102,
938, 1009, 1082, 1092, 1093, 1164	426, 436, 438, 676, 687, 689,
<code>\@glo@type</code> . . . 11, 18, 80, 98–100,	932, 943, 945, 1087, 1098, 1100
102, 103, 180, 181, 198, 204,	

<code>\@glo@uservaccess</code>	<code>\@gls@adjustmode</code>
349–351, 550, 551, 553, 804–806	... 181, 181 , 481, 733, 988, 1143
<code>\@glo@uservi</code>	<code>\@gls@after</code> 21, 393, 394, 643, 898, 1052
426, 436, 438, 676, 687, 689,	<code>\@gls@assign@symbol@field</code>
932, 943, 945, 1087, 1098, 1100	25
<code>\@glo@userviaccess</code>	<code>\@gls@assign@symbolplural@field</code> 25
349–351, 550, 551, 553, 804–806	<code>\@gls@automake</code>
<code>\@glodesc</code> ... 102, 439, 689, 946, 1100	199, 490, 491, 494, 742, 743,
<code>\@glolist@</code> .. 100, 437, 688, 944, 1098	746, 997, 998, 1001, 1153, 1156
<code>\@gloname</code> ... 102, 438, 689, 945, 1100	<code>\@gls@automake@immediate</code>
<code>\@glossary@default@style</code> 195 , 197, 491–
... 11, 11 , 13, 218,	493, 743–745, 998, 999, 1154, 1155
246, 274, 388, 389, 504, 518,	<code>\@gls@automake@types</code>
546, 638, 639, 756, 770, 799,	197 , 199, 999, 1001, 1155, 1156
892, 893, 1011, 1024, 1039,	<code>\@gls@before</code> .. 21, 393, 643, 898, 1052
1040, 1047, 1048, 1167, 1180, 1195	<code>\@gls@between</code>
<code>\@glossaryentryfield</code> 104 , 105, 106,	... 283, 605, 606, 859, 1042, 1043
440, 441, 691, 947, 948, 1101, 1102	<code>\@gls@body</code>
<code>\@glossarysection</code>	172, 477, 728, 729, 983, 1138, 1139
410, 411, 660, 661, 916, 1070, 1071	<code>\@gls@checkactual</code>
<code>\@glossarystyle</code> 131, 135 , 191, 457, 461,
218, 219, 237, 504, 512, 756,	488, 708, 712, 740, 962, 965,
764, 1011, 1012, 1020, 1167, 1175	966, 995, 1116, 1120, 1121, 1151
<code>\@glossarysubentryfield</code>	<code>\@gls@checkbox</code>
... 104 , 105, 106,	... 131, 134 , 191, 457, 460,
440, 441, 691, 692, 947, 948, 1102	489, 708, 711, 740, 962, 964,
<code>\@gls</code> ... 140, 140 , 463, 715, 968, 1124	965, 995, 1116, 1119, 1120, 1151
<code>\@gls@</code>	<code>\@gls@checkedmkidx</code>
112, 114, 140, 140 ,	... 130–137,
341, 343, 444, 445, 463, 602,	190–192, 456–462, 487–489,
603, 695, 696, 715, 855, 856,	707–714, 739–741, 961–967,
952, 953, 968, 1106, 1108, 1124	994–996, 1116–1122, 1150–1152
<code>\@gls@@automake@immediate</code> . 197 ,	<code>\@gls@checkesc</code>
198, 493, 744, 745, 999, 1000, 1155	... 131, 136 ,
<code>\@gls@link</code>	192, 708, 741, 962, 995, 1117, 1151
125, 126, 453, 454, 704, 958, 1113	<code>\@gls@checkescactual</code> 131, 132 , 191,
<code>\@gls@Hcounter</code>	457–459, 488, 708–710, 740,
128,	961, 963, 995, 1116, 1118, 1150
129, 455, 706, 959, 960, 1114, 1115	<code>\@gls@checkeschar</code>
<code>\@gls@ReturnAfterFi</code> 131, 133 , 192, 457,
... 249, 519, 520, 771, 1026, 1182	459, 489, 708, 710, 740, 962–
<code>\@gls@access@display</code> .. 355 , 554, 808	964, 995, 1116, 1118, 1119, 1151
<code>\@gls@actualchar</code> 106, 133, 135, 188,	<code>\@gls@checkesclevel</code> 131, 133 , 192,
441, 458, 459, 461, 486, 588,	459, 460, 708, 710, 711, 741,
691, 692, 709, 710, 712, 737,	962, 964, 995, 1117, 1119, 1151
738, 841, 947, 948, 963, 965,	<code>\@gls@checkescquote</code> ... 131, 132 ,
966, 992, 1102, 1118, 1120, 1148	191–193, 457, 458, 488–490,
<code>\@gls@addpredefinedattributes</code> .	708, 709, 740, 741, 961–963,
... 60 , 184, 193, 414, 483,	995, 996, 1116–1118, 1150–1152
490, 664, 735, 742, 919, 920,	<code>\@gls@checklevel</code> 131, 135 , 192, 457,
989, 997, 1074, 1075, 1145, 1152	460, 489, 708, 711, 712, 740,
	962, 965, 995, 1117, 1120, 1151

<code>\@gls@checkmkidxchars</code>	. 105, 129, 130, 191, 200, 212, 214, 440, 456, 457, 488, 494, 500, 502, 585, 691, 707, 708, 740, 746, 752, 753, 838, 947, 960, 961, 995, 1001, 1008, 1009, 1102, 1115, 1116, 1150, 1157, 1163, 1165	926, 927, 930, 933, 934, 943– 946, 1039, 1081, 1082, 1085, 1088, 1089, 1097, 1099, 1100, 1195
<code>\@gls@checkquote</code> 131, 131, 190, 191, 457, 458, 487, 488, 708, 709, 739, 740, 961, 962, 994, 995, 1116, 1117, 1149, 1150	<code>\@gls@defaulttype</code> . 74, 74, 928, 1082 <code>\@gls@deffile</code> 87–89, 429, 430, 679, 680, 935, 936, 1090, 1091
<code>\@gls@classI</code> 185, 483, 484, 735, 990, 1145, 1146	<code>\@gls@define@glossaryentrycounter</code> 14, 14, 45, 238, 239, 390, 406, 513, 514, 639, 640, 656, 765, 766, 893, 894, 911, 1020, 1021, 1048, 1066, 1176, 1177
<code>\@gls@classII</code>	185, 484, 735, 990, 1146	<code>\@gls@define@glossarysubentrycounter</code> 15, 45, 238, 239, 390, 406, 513, 514, 640, 656, 765, 766, 894, 912, 1020, 1021, 1049, 1066, 1176, 1177
<code>\@gls@codepage</code> 65, 220, 221, 417, 505, 667, 757, 923, 1013, 1078, 1168	<code>\@gls@defsort</code> 16–19, 103, 391, 392, 439, 640–642, 690, 894–896, 946, 1049–1051, 1100
<code>\@gls@counter</code> 123, 127–129, 180, 181, 204, 212, 213, 453–456, 481, 497, 501, 585, 704–706, 733, 749, 753, 838, 957, 959, 960, 988, 1004, 1008, 1009, 1112, 1114, 1115, 1143, 1160, 1164, 1165	<code>\@gls@defsortcount</code> ... 16–19, 76, 391, 392, 423, 640–642, 673, 894–896, 929, 1049–1051, 1083
<code>\@gls@counterwithin</code> 14, 15, 390, 639, 640, 893, 894, 1048, 1049	<code>\@gls@deprecated@acr@warning</code> 272, 787, 789, 791, 793, 795, 797, 1039, 1195
<code>\@gls@ctr</code> 58, 412, 663, 918, 1073	<code>\@gls@do@acronymsdef</code> 20, 20, 45, 46, 77, 392, 393, 406, 423, 642, 643, 656, 673, 897, 911, 912, 929, 1051, 1052, 1066, 1084
<code>\@gls@currentlettergroup</code> 232, 234, 509, 511, 761, 763, 1017, 1018, 1172, 1174	<code>\@gls@do@automake@err</code> 36, 37, 38, 907, 908, 1062, 1063
<code>\@gls@debugfalse</code>	7, 386, 636, 890, 1044	<code>\@gls@do@glsunset</code> 124, 124, 141–145, 957, 969–971, 1112, 1124–1126
<code>\@gls@debugtrue</code> 7, 386, 635, 636, 889, 890, 1044, 1045	<code>\@gls@do@indexdef</code> 41, 45, 46, 77, 405, 406, 423, 655, 656, 673, 910–912, 929, 1065, 1066, 1084
<code>\@gls@declareoption</code> 11, 12, 13, 20, 23, 24, 30, 33– 36, 38, 40, 41, 388, 389, 392– 395, 399, 401, 402, 404, 405, 638, 639, 642–645, 649, 651, 652, 654, 655, 892, 893, 896– 899, 903, 905–908, 910, 1047, 1048, 1051–1054, 1058, 1060–1065	<code>\@gls@do@numbersdef</code> 41, 45, 46, 77, 404–406, 423, 655, 656, 673, 910–912, 929, 1065, 1066, 1084
<code>\@gls@default</code> 117, 446, 697, 954, 1108, 1109	<code>\@gls@do@symbolsdef</code> . 40, 77, 404, 423, 654, 673, 910, 929, 1064, 1084
<code>\@gls@default@value</code> 71–73, 79, 84, 85, 98, 99, 101, 102, 272, 339, 420, 421, 424, 427, 428, 436–439, 541, 600, 601, 670, 671, 674, 677, 678, 686–689, 794, 854,	<code>\@gls@do@symbolssdef</code> 45, 46, 406, 656, 911, 912, 1066
		<code>\@gls@doautomake</code> 37, 37, 38, 197, 199, 403, 493, 494, 653, 745, 746, 908, 1000, 1001, 1063, 1155, 1156
		<code>\@gls@docheckquotedef</code>

..... 190–193, 487–	\@gls@fetchfield ... 73, <u>90</u> , 421,
490, 739–742, 994–996, 1149–1152	430, 671, 681, 927, 937, 1082, 1092
\@gls@docloadedfalse	\@gls@field@link
..... 6, 385, 635, 889, 1044	93,
\@gls@docloadedtrue	146, <u>146</u> , 147–161, 366–371,
..... 6, 385, 635, 889, 1044	432, 433, 466–472, 566–569,
\@gls@dodolistparser	683, 718–723, 819–822, 939,
. 199, 494, 745, 1000, 1001, 1156	971–977, 1093, 1094, 1127–1133
\@gls@doentrycounterdef	\@gls@fieldaccess@display
. 45, 46, 406, 656, 911, 912, 1066	. <u>355</u> , 356–358, 555–557, 808–810
\@gls@doentrydef . 452, 453, 703, 704	\@gls@firsttok
\@gls@dolast 232, 510, 762, 1017, 1173
. 215, 502, 754, 1010, 1165, 1166	\@gls@fixbraces 104, <u>214</u> , 439, 502,
\@gls@donext	690, 754, 946, 1009, 1101, 1165
. 215, 502, 754, 1010, 1165, 1166	\@gls@forbidtexext
\@gls@donext@def <u>74</u> , 76, 422, 672, 673, 928, 1083
179, 180, 481, 732, 733, 987, 1143	\@gls@fulllist ... 35, 652, 907, 1061
\@gls@dosubentrycounterdef	\@gls@get@counterprefix
..... 45, 46, 406, 656, 912, 1066 213, 501, 753, 1009, 1164
\@gls@dothiswrite	\@gls@getbody 172, 477, 728, 983, 1138
491,	\@gls@getcounter
492, 743, 744, 997–999, 1153–1155 <u>77</u> , 423, 673, 929, 1084
\@gls@elem	\@gls@getcounterprefix ... 213,
605, 858, 1042	<u>213</u> , 499–501, 751–753, 1009, 1164
\@gls@enablesavenonumberlist ..	\@gls@getgrouptitle 200, 244, <u>244</u> ,
..... <u>82</u> , 88, 425,	283, 494, 517, 606, 746, 769,
429, 676, 680, 932, 936, 1086, 1090	859, 1002, 1023, 1043, 1157, 1179
\@gls@encapchar	\@gls@glossary
133, 134, 188, 213, 214, 459,	205, <u>205</u> ,
460, 486, 501, 502, 585, 588,	206, 498, 750, 1005, 1160, 1161
710, 711, 737, 738, 753, 754,	\@gls@glslink@hyper@update@hook
838, 841, 964, 965, 992, 1008, 123, <u>123</u> , 957, 1112
1009, 1118–1120, 1148, 1164, 1165	\@gls@gobbleopt 75, 422, 672, 928, 1083
\@gls@entry@count 113, 114, <u>114</u> ,	\@gls@grptitle
444, 445, 695, 696, 952, 953, 1107	244,
\@gls@entry@field	280, 283, 517, 605, 606, 769,
91, 92,	858, 859, 1023, 1042, 1043, 1179
112, 138, <u>169</u> , 170–179, 351–	\@gls@hyp@opt
353, 431, 432, 443, 476–480,	. 93, 114–116, <u>124</u> , 125, 140–
553, 554, 682, 694, 728–732,	168, 253–255, 341–344, 432,
806, 807, 938, 951, 982–986,	433, 445, 453, 463–476, 520,
1092, 1093, 1106, 1123, 1137–1142	521, 602–604, 683, 696, 704,
\@gls@escbsdq <u>130</u> , 131, 189,	715–727, 772, 773, 855–857,
456, 457, 487, 589, 707, 708,	939, 953, 958, 968–981, 1027,
738, 739, 842, 961, 993, 1116, 1149	1028, 1093, 1094, 1107, 1108,
\@gls@expand@fields	1112, 1113, 1124–1137, 1183, 1184
85,	\@gls@hyp@opt@cs
<u>85</u> , 86, 428, 678, 934, 1088, 1089	124, 125, 453, 704, 958, 1112, 1113
\@gls@expandonce	\@gls@hypergroup
..... 86, 428, 678, 934, 1089 282, <u>283</u> , 605, 858, 1042
\@gls@extramakeindexopts	\@gls@if@measuring
..... <u>193</u> , 198, 490, 493, <u>106</u> , 107, 948, 1102, 1103
742, 745, 996, 1000, 1152, 1156	
\@gls@fetch@lettergrp . 225, 226, 233	

<code>\@gls@ifinlist</code>	<code>\@gls@link@opts</code>
. 58, <u>58</u> , 412, 662, 663, 918, 1073	534, 539, 705, 786, 792, 959, 1114
<code>\@gls@ifnotmeasuring</code>	<code>\@gls@list</code> ..
441, 692	605, 606, 858, 859, 1042
<code>\@gls@igtype</code> ..	<code>\@gls@listsuffix</code>
78, 424, 674, 930, 1084 58, 412, 662, 918, 1073
<code>\@gls@increment@currcount</code>	<code>\@gls@loadlist</code>
..... 112, <u>113</u> , 443,	.. 13, <u>13</u> , 273, 389, 546, 639,
444, 694, 695, 951, 952, 1106, 1107	799, 893, 1039, 1047, 1048, 1195
<code>\@gls@indexdef</code>	<code>\@gls@loadlong</code>
.... 41, <u>41</u> , 405, 655, 910, 1065	12,
<code>\@gls@initnonumberlist</code>	<u>12</u> , 13, 273, 389, 546, 638, 639,
... 82, <u>82</u> , 99, 425, 436, 675,	799, 893, 1039, 1047, 1048, 1195
676, 687, 931, 932, 943, 1086, 1097	<code>\@gls@loadshellesc</code>
<code>\@gls@islistofacronyms</code>	43–45
..... 21, 393, 643, 897, 1052	<code>\@gls@loadsuper</code>
<code>\@gls@keylist</code>	12, <u>12</u> , 13, 273, 389, 546, 639,
382, 578, 831	799, 893, 1039, 1047, 1048, 1195
<code>\@gls@keymap</code> ...	<code>\@gls@loadtree</code> 13, <u>13</u> , 273, 389, 546,
82, 89, <u>89</u> , 90–92,	639, 799, 893, 1039, 1048, 1195
339, 349, 426, 429–431, 551,	<code>\@gls@local@increment@currcount</code>
600, 676, 680–682, 804, 853, 112, <u>113</u> , 443,
932, 936–938, 1087, 1091–1093	444, 694, 695, 951, 952, 1106, 1107
<code>\@gls@label</code> ...	<code>\@gls@loclist</code>
200, 204, 209, 212,	201,
213, 494, 495, 497, 500, 501,	202, 233, 234, 495, 496, 510,
746, 749, 751–753, 1002, 1004,	511, 747, 748, 762, 763, 1002,
1007, 1008, 1157, 1160, 1163, 1164	1003, 1018, 1158, 1159, 1173, 1174
<code>\@gls@labelsfile</code>	<code>\@gls@map</code>
35, 36, 402, 652, 907, 1061, 1062	89, 90, 429,
<code>\@gls@langmod</code> ..	430, 680, 681, 936, 937, 1091, 1092
33, 195, 196, 220,	<code>\@gls@missinglang@warn</code>
491, 492, 743, 744, 905, 997, 23, <u>23</u> , 50, 395,
998, 1012, 1060, 1153, 1154, 1168	408, 644, 658, 899, 914, 1053, 1068
<code>\@gls@levelchar</code>	<code>\@gls@missingnumberlist</code>
.... 106, 134, 135, 188, 441, <u>84</u> , 102, 427,
459, 460, 486, 588, 692, 710,	438, 677, 689, 933, 945, 1087, 1100
711, 737, 738, 841, 948, 964,	<code>\@gls@noaccess</code>
965, 992, 1102, 1119, 1120, 1148
<code>\@gls@link</code> 125, <u>127</u> , 141–146, 162–	349, 355, 551, 554, 555, 804, 808
169, 364–366, 454, 463–466,	<code>\@gls@noexpand@field</code>
472–476, 564, 565, 704, 705, <u>84</u>
715–718, 723–728, 817–819,	<code>\@gls@noexpand@fields</code> <u>84</u> , 86, 427,
958, 959, 969–971, 977–982,	428, 677, 678, 933, 934, 1088, 1089
1113, 1114, 1124–1127, 1133–1137	<code>\@gls@nohyperlist</code>
<code>\@gls@link@checkfirsthyper</code> 22, <u>22</u> , 78, 126, 394,
..... <u>126</u> , 140–145,	423, 424, 454, 644, 674, 705,
454, 463–466, 705, 715–717,	898, 930, 959, 1053, 1084, 1113
958, 968–971, 1113, 1124–1126	<code>\@gls@noidx@do</code>
<code>\@gls@link@label</code>	232, <u>233</u> ,
127, 454,	510, 761, 762, 1017, 1018, 1173
534, 539, 705, 786, 792, 959, 1114	<code>\@gls@noidx@getgrouptitle</code>
<code>\@gls@link@nocheckfirsthyper</code> 200, <u>244</u> , 494, 517,
..... 146, <u>146</u> , 161–169,	746, 769, 1002, 1023, 1157, 1179
466, 472–476, 718, 723–727,	<code>\@gls@noidx@sanitizesort</code>
971, 977–982, 1127, 1133–1137 25, 203, 396, 496,
	645, 748, 900, 1003, 1054, 1159

<code>\@gls@noidx@setsanitizesort</code>	28, 28, 203, 398, 496, 647, 648, 748, 902, 1003, 1056, 1057, 1159	<code>\@gls@prevlevel</code>	.. 312, 313, 334– 336, 595, 596, 621, 622, 633, 634, 848, 849, 874, 875, 887, 888
<code>\@gls@noidx@loclist@finalsep</code> 202, 496, 747, 748, 1003, 1158	<code>\@gls@provide@newglossary</code>	74, 76, 422, 672, 673, 927–929, 1082, 1083
<code>\@gls@noidx@loclist@prev</code>	202, 234, 235, 496, 511, 512, 748, 763, 764, 1003, 1019, 1158, 1174, 1175	<code>\@gls@quotechar</code> 132–135, 188, 190–192, 457– 461, 486–489, 588, 709–712, 737–741, 841, 962–966, 992, 994, 996, 1117–1120, 1148–1152
<code>\@gls@noidx@loclist@sep</code> 202, 234, 235, 495, 496, 511, 512, 747, 763, 764, 1003, 1019, 1158, 1174, 1175	<code>\@gls@reference</code> 200, 201, 204, 235, 494, 495, 497, 512, 746, 747, 749, 764, 1002, 1004, 1019, 1157, 1160, 1175
<code>\@gls@noist@toolate</code>	188, 190, 193, 992, 993, 997, 1148, 1149, 1152	<code>\@gls@removedoption</code> 27, 32, 38, 39, 901, 905, 909, 1056, 1059, 1060, 1063, 1064
<code>\@gls@noref@warn</code>	. 200, 203, 232, 494, 497, 510, 746, 749, 762, 1001, 1004, 1017, 1157, 1159, 1173	<code>\@gls@removespaces</code> 248, 249, 519, 771, 1025, 1026, 1181, 1182
<code>\@gls@notranslatorhook</code> 29	<code>\@gls@renewglossary</code> 194, 206, 490, 498, 742, 750, 997, 1005, 1152, 1161
<code>\@gls@numberlink</code> 248, 519, 771, 1025, 1181	<code>\@gls@replacementtext</code> 354, 355, 554, 807, 808
<code>\@gls@numbersdef</code> 41, 41, 404, 405, 655, 910, 1065	<code>\@gls@requested@languages</code> 31, 31, 48, 904, 913, 1059, 1068
<code>\@gls@numlist@lastsep</code> 179, 180, 481, 732, 733, 987, 1143	<code>\@gls@rest</code>	172, 477, 728, 729, 983, 1139
<code>\@gls@numlist@nextsep</code> 179, 180, 481, 732, 733, 987, 1143	<code>\@gls@restore@glsllocal</code> 124, 128, 957, 959, 1112, 1114
<code>\@gls@numlist@sep</code> 179, 180, 481, 732, 733, 987, 1143	<code>\@gls@restoreat</code>	88, 429, 680, 936, 1090
<code>\@gls@old@chapter</code> 46, 406, 656, 912, 1067	<code>\@gls@roman</code> 61, 62, 414, 415, 586, 665, 839, 920, 921, 1075
<code>\@gls@oldnewglossaryentryposthook</code> 350, 551, 805	<code>\@gls@run@output@dir</code>	. 37, 38, 42, 44
<code>\@gls@oldnewglossaryentryprehook</code> 349, 350, 551, 804	<code>\@gls@run@output@file</code>	43, 44, 195, 196
<code>\@gls@onlypremakeg</code> 46, 46, 406, 656, 657, 912, 1067	<code>\@gls@run@unrestricted@shell</code> 42, 43, 195–197
<code>\@gls@order</code> 194–196, 491, 492, 742–744, 997–999, 1153–1155	<code>\@gls@runshell</code>	42, 42, 43, 44, 195, 196
<code>\@gls@org@LT@output</code>	... 301, 614, 868	<code>\@gls@sanitized@tmp</code> 130, 456, 457, 707, 961, 1116
<code>\@gls@org@glsnoidxdisplayloc</code> 202, 496, 748, 1003, 1159	<code>\@gls@sanitizedesc</code> 24, 395, 400, 645, 650
<code>\@gls@org@glssseeformat</code> 202, 496, 748, 1003, 1159	<code>\@gls@sanitizename</code> 25, 395, 645, 899, 1054
<code>\@gls@override@glossary</code>	... 39, 40, 403, 404, 654, 909, 910, 1064	<code>\@gls@sanitizesort</code>	.. 16, 25, 391, 395, 640, 645, 894, 899, 1049, 1054
<code>\@gls@patchtabularx</code> 108, 441, 692, 949, 1104	<code>\@gls@sanitizesymbol</code> 25, 395, 400, 401, 645, 650, 899, 1054
<code>\@gls@preglossaryhook</code> 218, 219, 503, 504, 755, 756, 1011, 1012, 1167, 1168	<code>\@gls@save@glsllocal</code> 123, 124, 127, 957, 959, 1112, 1114

<code>\@gls@saveentrycounter</code>	<code>\@gls@thislabel</code> 70, 215, 231, 419,
. 127, <u>128</u> , 181, 455, 481,	502, 509, 670, 754, 761, 925,
705, 706, 733, 959, 988, 1114, 1143	1010, 1016, 1080, 1165, 1166, 1172
<code>\@gls@savenonumberlist</code>	<code>\@gls@thislang</code> 48, 913, 1068
. 81, 82, <u>82</u> , 203, 425, 675,	<code>\@gls@thislist</code>
676, 931, 932, 1004, 1086, 1159	179, 180, 481, 732, 733, 987, 1143
<code>\@gls@see@noindex</code>	<code>\@gls@thisloc</code> 213, 501, 753, 1009, 1164
. 9, <u>9</u> , 81, 387, 388,	<code>\@gls@thisval</code> . 90, 430, 681, 937, 1092
425, 637, 675, 891, 931, 1046, 1086	<code>\@gls@title</code> . . . 54, 410, 660, 916, 1070
<code>\@gls@setacrstyle</code>	<code>\@gls@tmp</code> 8,
. 32, 45, 400, 401, 405, 406,	18, 49, 61, 63, 86, 130, 206,
650, 656, 904, 911, 912, 1059, 1066	248, 249, 387, 392, 407, 415,
<code>\@gls@setacounter</code>	428, 456, 457, 498, 605, 606,
76, <u>77</u> , 423, 673, 929, 1083, 1084	636, 641, 658, 665, 666, 678,
<code>\@gls@setdefault@glslink@opts</code> .	707, 750, 859, 891, 896, 913,
. <u>126</u> , 127, 454, 705, 959, 1114	920, 921, 934, 961, 1005, 1026,
<code>\@gls@setsort</code> 16–19,	1042, 1043, 1045, 1050, 1068,
127, 181, 391, 392, 455, 481,	1075, 1076, 1089, 1116, 1161, 1181
640–642, 705, 733, 895, 896,	<code>\@gls@tmpb</code> . . . 131, <u>131</u> , 132–137,
959, 988, 1049–1051, 1114, 1143	190–192, 457–462, 487–489,
<code>\@gls@setupshortcuts</code>	708–714, 739–741, 962–967,
45, 405, 406, 656, 911, 912, 1066	994, 996, 1117–1122, 1150–1152
<code>\@gls@setupsort@clear</code> . <u>18</u> , 896, 1051	<code>\@gls@toc</code> 56, <u>57</u> ,
<code>\@gls@setupsort@def</code>	411, 412, 661, 662, 917, 1071, 1072
. <u>17</u> , 391, 641, 895, 1050	<code>\@gls@type</code> . . . 197, 199, 257, 333,
<code>\@gls@setupsort@none</code>	493, 494, 523, 533, 535, 537,
. <u>18</u> , 392, 642, 896, 1050	538, 540, 543, 544, 546, 632,
<code>\@gls@setupsort@standard</code>	745, 746, 775, 785, 788, 789,
<u>16</u> , 390, 391, 640, 894, 895, 1049	791, 793, 796, 797, 799, 886,
<code>\@gls@setupsort@use</code>	999, 1001, 1030, 1155, 1156, 1186
. <u>17</u> , 391, 641, 895, 1050	<code>\@gls@updatechecked</code> . . . 130, 131,
<code>\@gls@sort</code>	<u>131</u> , 191, 192, 456, 457, 488,
225, 226, 233, 511, 763, 1018, 1174	489, 707, 708, 740, 741, 961,
<code>\@gls@sort@A</code> 228,	962, 995, 1116, 1117, 1150, 1151
229, 507, 759, 1014, 1015, 1170	<code>\@gls@usetranslator</code> <u>29</u> , 30, 31, 49,
<code>\@gls@sort@B</code> 228,	399, 400, 407, 648, 649, 657,
229, 507, 759, 1014, 1015, 1170	903, 904, 913, 1057–1059, 1068
<code>\@gls@startwithexpandonce</code>	<code>\@gls@value</code> 84,
. 85, <u>86</u> , 428, 678, 934, 1089	85, 175, 427, 428, 478, 677,
<code>\@gls@storenonumberlist</code>	678, 730, 933, 934, 985, 1088, 1140
. 82, 102, 425, 438,	<code>\@gls@warnonglossdefined</code>
675, 676, 689, 932, 945, 1086, 1100 24, <u>24</u> , 216, 395, 503,
<code>\@gls@symbolsdef</code>	645, 755, 899, 1010, 1054, 1166
. 40, <u>40</u> , 404, 654, 910, 1064	<code>\@gls@warnontheGLOSSdefined</code>
<code>\@gls@this</code> . 207, 499, 751, 1006, 1161 24, <u>24</u> , 241, 395, 515,
<code>\@gls@thisHloc</code>	645, 767, 899, 1022, 1054, 1178
. 213, 501, 753, 1009, 1164	<code>\@gls@write@entrycounts</code>
<code>\@gls@thisentryname</code>	113, <u>114</u> , 444, 445, 695, 952, 1107
. 35, 36, 652, 907, 1061, 1062	<code>\@gls@writedef</code> 88, <u>89</u> ,
<code>\@gls@thisfield</code> 73, 421, 671, 927, 1082	429, 679, 680, 935, 936, 1090, 1091

<code>\@gls@writeisthook</code>	<code>\@glsdesc</code> ... 151, 468, 720, 973, 1129
..... 188, 190, 190, 486,	<code>\@glsdesc@</code> 151,
487, 737, 739, 992–994, 1148, 1149	368, 468, 567, 720, 820, 973, 1129
<code>\@gls@x</code> 248, 249, 771, 1026, 1181, 1182	<code>\@glsdescplural</code>
<code>\@gls@xdy@Lclass@Alpha-page-numbers</code>	... 152, 468, 720, 973, 974, 1129
..... 62	<code>\@glsdescplural@</code> 152,
<code>\@gls@xdy@Lclass@Appendix-page-numbers</code>	368, 468, 567, 720, 820, 974, 1129
..... 62	<code>\@glsdisp</code>
<code>\@gls@xdy@Lclass@Roman-page-numbers</code>	145, 145, 146, 466, 717, 971, 1126
..... 62	<code>\@glsentry</code> 35,
<code>\@gls@xdy@Lclass@alpha-page-numbers</code>	36, 110, 111, 114, 402, 442,
..... 62	443, 445, 652, 693–696, 907,
<code>\@gls@xdy@Lclass@arabic-page-numbers</code>	950–953, 1061, 1062, 1105, 1107
..... 62	<code>\@glsentrytitlecase</code>
<code>\@gls@xdy@Lclass@arabic-section-numbers</code>	175, 175, 478, 479, 730, 985, 1140
..... 62	<code>\@glsfirst</code> 147, 467, 718, 971, 972, 1127
<code>\@gls@xdy@Lclass@roman-page-numbers</code>	<code>\@glsfirst@</code> 147,
..... 61	367, 467, 566, 718, 819, 972, 1127
<code>\@gls@xdy@locationlist</code>	<code>\@glsfirstletter</code>
. 61, 185, 414, 483, 484, 664,	183, 183, 482, 734, 988, 989, 1144
735, 920, 990, 1075, 1145, 1146	<code>\@glsfirstplural</code>
<code>\@gls@xdy@checkbackslash</code> 149, 467, 719, 972, 1128
..... 130, 137, 456, 462,	<code>\@glsfirstplural@</code> 149, 367,
707, 713, 714, 961, 967, 1116, 1122	367, 467, 566, 719, 819, 972, 1128
<code>\@gls@xdy@checkquote</code>	<code>\@glsnumber</code> 247, 247,
130, 136, 456, 461, 707, 713,	519, 770, 771, 1025, 1180, 1181
961, 966, 967, 1116, 1121, 1122	<code>\@glsisacronymlistfalse</code>
<code>\@gls@xref</code> 22, 394, 643, 898, 1052
.. 214, 502, 753, 754, 1009, 1165	<code>\@glsisacronymlisttrue</code>
<code>\@gls@Alphacompositor</code> 22, 394, 643, 898, 1052
... 52, 52, 62, 409, 415, 586,	<code>\@glslink</code> 107, 108, 127, 139, 139,
659, 665, 840, 915, 921, 1070, 1076	140, 180, 280, 455, 462, 463,
<code>\@gls@Hlocref</code> .. 209, 212, 499, 500,	481, 605, 706, 714, 715, 733,
751, 752, 1007, 1008, 1162, 1163	858, 948, 949, 959, 968, 988,
<code>\@gls@acronymlists</code> 21,	1042, 1103, 1114, 1123, 1124, 1143
21, 22, 67, 256, 257, 278, 393,	<code>\@glslocalreset</code> 109, 110, 112, 442,
394, 418, 522, 523, 533, 535–	443, 692–694, 949–951, 1104–1106
538, 540, 542–544, 546, 548,	<code>\@glslocalunset</code> 109, 109, 112, 442,
643, 668, 774, 775, 785, 787–	443, 693, 694, 950, 951, 1105, 1106
789, 791, 793, 795–797, 799,	<code>\@glslocref</code>
801, 897, 898, 924, 1029, 1030,	204, 209, 211–213, 497, 499–
1041, 1052, 1079, 1184–1186, 1197	501, 585, 749, 751–753, 838,
<code>\@gls@addkey</code> 91, 92, 431, 682, 938, 1093	1004, 1007, 1008, 1160, 1162–1164
<code>\@gls@addstoragekey</code>	<code>\@glsminrange</code> 184, 184,
91, 431, 681, 682, 937, 938, 1092	185, 483, 484, 586, 587, 734–
<code>\@gls@addxdyattribute</code>	736, 839, 840, 989, 990, 1145, 1146
59, 60, 413, 663, 919, 1073, 1074	<code>\@glsname</code> ... 150, 468, 719, 973, 1128
<code>\@gls@defaultplural</code> 84, 427, 677	<code>\@glsname@</code> 150,
<code>\@gls@defaultsort</code> ... 16, 84, 391,	368, 468, 566, 719, 820, 973, 1128
427, 640, 677, 894, 933, 1049, 1088	

<code>\@glsnavhypertarget</code>	<code>\@glssymbolplural</code>
. 282, <u>282</u> , 605, 858, 1042 154, 469, 721, 974, 1130
<code>\@glsnextpages</code> 219, <u>238</u> , 504, 514,	<code>\@glssymbolplural@</code> 154,
756, 765, 1012, 1021, 1167, 1177	<u>369</u> , 469, 567, 721, 820, 974, 1130
<code>\@glsnodesc</code> <u>83</u> , 98,	<code>\@glstarget</code> 139, <u>139</u> , 140, 241, 282,
100, 102, 426, 436, 439, 677,	463, 515, 605, 714, 715, 767,
686, 687, 689, 933, 942, 943,	858, 968, 1022, 1042, 1124, 1178
945, 946, 1087, 1097, 1098, 1100	<code>\@glstext</code> . . . 146, 466, 718, 971, 1127
<code>\@glsnoname</code>	<code>\@glstext@</code> 146,
<u>83</u> , 98, 101, 102, 426, 436–438,	<u>366</u> , 466, 566, 718, 819, 971, 1127
676, 686, 688, 689, 932, 942,	<code>\@glsunset</code> . . . 109, <u>109</u> , 112, 442,
944, 945, 1087, 1097, 1099, 1100	443, 693, 694, 950, 951, 1104–1106
<code>\@glsnonextpages</code> 219, <u>238</u> , 504, 513,	<code>\@glsuseri</code> . . 155, 469, 721, 975, 1130
756, 765, 1012, 1021, 1167, 1176	<code>\@glsuseri@</code> 155,
<code>\@glsnumberformat</code>	<u>369</u> , 469, 568, 721, 821, 975, 1130
123, 127, 180, 181, 204, 212,	<code>\@glsuserii</code> . . 156, 470, 721, 975, 1131
213, 453, 454, 481, 497, 501,	<code>\@glsuserii@</code> 156,
585, 704, 705, 733, 749, 752,	<u>370</u> , 470, 568, 721, 821, 975, 1131
753, 838, 957, 959, 988, 1004,	<code>\@glsuseriii</code>
1008, 1112, 1114, 1143, 1160, 1164	. . . 157, 470, 722, 975, 976, 1131
<code>\@glsopenfile</code>	<code>\@glsuseriii@</code> 157,
194, <u>194</u> , 205, 490, 498, 742,	<u>370</u> , 470, 568, 722, 821, 976, 1131
749, 997, 1005, 1152, 1153, 1160	<code>\@glsuseriv</code>
<code>\@glsorder</code> <u>33</u> , 198, 401, 493,	. . 158, 470, 722, 976, 1131, 1132
650, 745, 905, 1000, 1060, 1156	<code>\@glsuseriv@</code> 158,
<code>\@glspl</code> . . 142, 143, 464, 716, 969, 1125	<u>370</u> , 471, 568, 722, 821, 976, 1132
<code>\@glspl@</code> 113, 115, 143, <u>143</u> ,	<code>\@glsuserv</code> . . 159, 471, 722, 976, 1132
342–344, 444, 445, 464, 602–	<code>\@glsuserv@</code> 159,
604, 695, 696, 716, 856, 857,	<u>371</u> , 471, 568, 723, 822, 976, 1132
952, 953, 969, 1107, 1108, 1125	<code>\@glsuservi</code> . . 160, 471, 723, 977, 1132
<code>\@glsplural</code>	<code>\@glsuservi@</code> 160,
. . 148, 467, 719, 972, 1127, 1128	<u>371</u> , 471, 569, 723, 822, 977, 1132
<code>\@glsplural@</code> 148,	<code>\@glswidestname</code> 333, <u>333</u> , 334, 335,
<u>367</u> , 467, 566, 719, 819, 972, 1128	595, 596, 633, 634, 849, 886, 887
<code>\@glsprefix@record@hook</code> . . 341,	<code>\@glswritefiles</code> 38, <u>204</u> , 403, 497,
<u>341</u> , 342–345, 602–604, 855–858	653, 749, 908, 1004, 1063, 1160
<code>\@glsreset</code> . . . 108, <u>110</u> , 112, 441–	<code>\@glsxtr@doaccsupp</code> 346, 548, 802
443, 692–694, 949–951, 1104–1106	<code>\@glsxtr@record</code> 341, 602, 855
<code>\@glssee</code> 104, 214, 439, 502,	<code>\@gobble</code> . 7, 16–18, 34–36, 89, 107,
690, 754, 946, 1010, 1101, 1165	130, 182, 183, 186, 198, 200,
<code>\@glsshowaccsupp</code>	287, 386, 391, 392, 401, 402,
. 7, <u>9</u> , 347, 386, 387, 549,	429, 430, 441, 456, 482, 484,
636, 637, 802, 890, 891, 1044, 1045	485, 493, 494, 584, 587, 588,
<code>\@glsshowtarget</code>	635, 636, 640–642, 651, 652,
. 7, 8, <u>8</u> , 386, 387, 462,	680, 681, 692, 707, 733, 734,
635, 636, 714, 889–891, 1044, 1045	736, 745, 746, 837, 840, 841,
<code>\@glssymbol</code>	861, 889, 890, 895, 896, 906,
. . 153, 469, 720, 974, 1129, 1130	907, 936, 937, 948, 961, 988,
<code>\@glssymbol@</code> 153,	991, 1000, 1001, 1044, 1049,
<u>369</u> , 469, 567, 720, 820, 974, 1130	

1050, 1061, 1062, 1091, 1103, 1116, 1144, 1146, 1147, 1156, 1157	\@mkboth
\@gobblethree 7, 386, 636, 890, 1044	55, 410, 411, 660, 661, 916, 1071
\@idxitem 328, 629, 883	\@one 107, 948, 1103
\@ifclassloaded 6, 14, 55, 385, 389, 410, 635, 639, 660, 889, 893, 916, 1044, 1048, 1071	\@newglossary 74, 76, <u>77</u> , 422, 423, 672, 673, 928, 929, 1082–1084
\@ifglossaryexists	\@newglossaryentry@defcounters 103, <u>104</u> , 111, 439, 440, 443, 690, 694, 946, 951, 1101, 1106
. 67, <u>68</u> , 216, 418, 503, 668, 669, 755, 924, 1010, 1079, 1166	\@newglossaryentryposthook 91, 92, 103, <u>104</u> , 339, 350, 431, 439, 440, 551, 601, 682, 690, 805, 854, 938, 946, 1092, 1093, 1101
\@ifl@t@r 27, 397, 647, 901, 1056	\@newglossaryentryprehook 91, 92, 97, 99, <u>104</u> , 339, 349, 350, 431, 435, 436, 440, 551, 600, 682, 686, 687, 690, 804, 854, 938, 942, 943, 946, 1092, 1093, 1097, 1098, 1101
\@ifnextchar 76, 124, 423, 453, 673, 704, 929, 958, 1083, 1112	\@nil 21, 104, 129–131, 172, 191, 192, 212, 214, 226, 232, 247–249, 394, 439, 456, 457, 477, 488, 489, 501, 502, 510, 511, 519, 585, 643, 690, 707, 708, 728, 740, 741, 752, 754, 762, 763, 770, 771, 838, 898, 946, 960–962, 983, 995, 1008, 1009, 1017, 1018, 1025, 1026, 1052, 1101, 1115–1117, 1138, 1150, 1151, 1164, 1165, 1173, 1174, 1180–1182
\@ifpackageloaded 6, 11, 12, 29–31, 49, 66, 107, 179, 190, 345, 385, 388, 399, 400, 407, 417, 441, 480, 487, 548, 635, 638, 648–650, 657, 668, 692, 732, 739, 801, 889, 892, 893, 903, 904, 913, 923, 948, 987, 994, 1044, 1047, 1057– 1059, 1068, 1078, 1103, 1142, 1149	\@nnil 21, 215, 394, 502, 643, 754, 898, 1010, 1052, 1165
\@ifstar 67, 75, 91, 124, 183, 250, 418, 422, 431, 453, 482, 520, 668, 672, 681, 682, 704, 734, 772, 924, 928, 937, 938, 958, 988, 989, 1026, 1079, 1083, 1092, 1093, 1112, 1144, 1182	\@no@makeglossaries 34, 35, 199, 201, <u>203</u> , 401, 402, 494, 495, 497, 651, 745, 747, 749, 906, 1001, 1002, 1004, 1060, 1061, 1156, 1158, 1159
\@ifundefined 48, 291, 302, 314, 321, 335, 407, 596, 605, 609, 615, 622, 626, 633, 634, 657, 849, 858, 863, 869, 875, 876, 879, 880, 887, 913, 1067	\@no@post@desc <u>51</u> , 409, 590, 659, 843, 914, 1069
\@ignored@glossaries 35, 77, 78, <u>78</u> , 402, 423, 424, 652, 673, 674, 907, 929, 930, 1061, 1084	\@nopostdesc <u>50</u> , 219, 408, 504, 659, 756, 914, 1012, 1069, 1167
\@input@ 220, 505, 756, 1012, 1168	\@onelevel@sanitize 8, 25, 35, 61, 89, 106, 130, 183, 187, 214, 216, 232, 249, 387, 396, 415, 429, 441, 456, 457, 482, 483, 485, 502, 503, 510, 586, 587, 636, 645, 646, 652, 665, 680, 691, 707, 734, 737, 753, 755, 762, 839,
\@istfilename <u>51</u> , 198, 409, 493, 659, 745, 915, 1000, 1069, 1156	
\@makecol 301, 614, 615, 868	
\@makeglossaries@warn@noprintglossary 199, <u>199</u> , 745, 746, 1000, 1001, 1156, 1157	
\@makeglossary	
. 34, 35, <u>194</u> , 198, 401, 402, 490, 493, 651, 652, 742, 745, 906, 997, 1000, 1061, 1152, 1156	
\@minus 287, 308, 327, 608, 618, 629, 861, 872, 883	

841, 891, 900, 907, 920, 936,
 947, 961, 989, 991, 1009, 1010,
 1017, 1026, 1045, 1054, 1062,
 1075, 1091, 1102, 1116, 1144,
 1145, 1147, 1165, 1166, 1173, 1181
 \@onlypreamble
 . . . 76, 87, 98, 113, 117, 199,
 203, 423, 429, 435, 444, 446,
 494, 496, 673, 679, 686, 695,
 697, 746, 748, 929, 935, 942,
 952, 954, 1001, 1004, 1084,
 1089, 1097, 1107, 1109, 1156, 1159
 \@onlypremakeg . 46, 51, 52, 59, 60,
 63, 77, 190, 406, 409, 413, 416,
 423, 487, 656, 659, 663, 664,
 666, 673, 739, 912, 915, 918,
 919, 922, 929, 994, 1067, 1069,
 1070, 1073, 1074, 1076, 1084, 1149
 \@org@Gls@acentryname 171, 983, 1138
 \@org@glossaryentrynumbers
 218, 219,
 504, 756, 1011, 1012, 1167, 1168
 \@org@Gls@assign@descplural
 533, 540,
 542, 544, 578, 581, 582, 785,
 792, 793, 795, 797, 832, 834, 835
 \@org@Gls@assign@firstpl
 533–536, 538,
 540, 542–544, 578–582, 784,
 785, 787–793, 795, 797, 831–835
 \@org@Gls@assign@plural 533–536,
 538, 540, 542–544, 578–582,
 785, 787–793, 795, 797, 832–835
 \@org@Gls@assign@symbolplural
 533–536, 538,
 542–544, 579, 580, 582, 785,
 787–791, 795, 797, 832, 833, 835
 \@org@Gls@numberformat
 179, 480, 732, 987, 1142
 \@org@newglossaryentryprehook
 97, 435, 686, 942, 1097
 \@outputpage 301, 614, 615, 868
 \@p@glossarysection . 54, 56, 410,
 411, 660, 661, 916, 917, 1070, 1071
 \@pgls 341, 341, 602, 855
 \@pgls@ 341, 341, 602, 855
 \@pglspl 342, 342, 602, 855
 \@pglspl@ 342, 342, 602, 855
 \@plus 287, 308,
 327, 608, 618, 629, 861, 872, 883
 \@print@glossary 217, 220, 503, 505,
 755, 756, 1011, 1012, 1166, 1168
 \@print@noidx@glossary
 217, 231, 503, 509,
 755, 761, 1011, 1017, 1167, 1172
 \@printgloss@checkexists
 217, 217, 218, 503,
 504, 755, 756, 1011, 1166, 1167
 \@printgloss@checkexists@allowignored
 217, 503, 755, 1011, 1166
 \@printgloss@checkexists@noignored
 216, 217,
 503, 755, 1010, 1011, 1166, 1167
 \@printgloss@setsort
 199, 201, 217, 218, 494, 495,
 503, 504, 745, 747, 755, 756,
 1001, 1002, 1011, 1156, 1158, 1167
 \@printglossary 217,
 218, 503, 755, 1011, 1166, 1167
 \@roman 61, 414, 586, 665, 839, 920, 1075
 \@secondofthree 124, 140, 141, 144,
 162, 164, 166, 168, 364, 453,
 463–465, 472–474, 476, 564,
 704, 715, 717, 724–727, 817,
 958, 968–970, 978–981, 1112,
 1124, 1126, 1133, 1134, 1136, 1137
 \@secondoftwo 26, 29–
 31, 49, 89, 90, 107, 139–142,
 145, 161–163, 165–167, 208,
 287, 337, 364–366, 396, 397,
 399, 400, 407, 429, 430, 463,
 464, 466, 472, 474, 475, 564,
 565, 583, 600, 646, 648–650,
 657, 680, 681, 714–717, 723,
 724, 726, 817–819, 836, 853,
 861, 900, 901, 903, 904, 913,
 936, 937, 948, 949, 968, 969,
 971, 977, 978, 980, 1007, 1055,
 1057–1059, 1068, 1091, 1092,
 1103, 1124–1126, 1133–1136, 1162
 \@set@glo@numformat
 129, 213, 456, 501, 585, 707,
 753, 838, 960, 1008, 1115, 1164
 \@sglsaddkey 91, 92, 431, 682, 938, 1093
 \@sglsaddstoragekey
 91, 431, 681, 937, 1092
 \@tabacckludge 26, 396, 646, 900, 1055
 \@text@composite@x
 26, 396, 646, 900, 1055

\@thirdofthree . . . 125, 142, 144,
163, 165, 167, 169, 365, 453,
464, 465, 472, 474–476, 564,
704, 716, 717, 724–726, 728,
818, 958, 969, 970, 978–980,
982, 1113, 1125, 1126, 1134–1137
\@this@attr
186, 484, 485, 736, 991, 1146, 1147
\@this@childlabel
. 227, 506, 758, 1013, 1169
\@this@counter 60, 413, 663, 919, 1074
\@this@ctr . . 186, 484, 736, 991, 1146
\@this@key . . . 90, 430, 681, 937, 1092
\@this@label
226, 227, 506, 757, 758, 1013, 1169
\@this@scs . 46, 406, 407, 657, 912, 1067
\@tmp 415, 586, 839
\@use@ption . . . 42, 405, 655, 911, 1066
\@warn@nomakeglossaries
. 197, 221, 493, 505,
744, 757, 999, 1013, 1155, 1168
\@wrglossary@pageformat
. . 208, 208, 499, 751, 1006, 1162
\@wrglossary@numberhook
. 208, 208, 211, 499, 500,
751, 752, 1006, 1008, 1162, 1163
\@write@glslabelsname
. 36, 36, 652, 907, 1062
\@xdy@main@language
. 33, 33, 401, 491, 492,
505, 651, 743, 744, 757, 905, 1060
\@xdy@attributelist 58, 59, 60, 186,
412, 413, 484, 662, 663, 736,
918, 919, 991, 1073, 1074, 1146
\@xdy@attributes 58,
59, 60, 185, 412, 413, 483, 584,
586, 662, 663, 735, 837, 839,
918, 919, 990, 1072–1074, 1145
\@xdy@counters 58, 60, 186,
412, 413, 484, 662, 663, 736,
918, 919, 991, 1073, 1074, 1146
\@xdy@crossrefhook . 63, 186, 415,
484, 666, 736, 921, 991, 1076, 1146
\@xdy@language . . 65, 220, 417, 505,
667, 757, 923, 1012, 1078, 1168
\@xdy@lettergroups 66,
66, 188, 418, 486, 588, 668,
737, 841, 923, 924, 992, 1078, 1148
\@xdy@locationclassorder
. . 63, 64, 186, 416, 484, 587,
666, 736, 840, 922, 991, 1077, 1146
\@xdy@locoref . 58, 59, 188, 412, 413,
485, 584, 588, 662, 663, 737,
837, 841, 918, 919, 992, 1073, 1147
\@xdy@numbergrouporder 66,
183, 183, 418, 482, 483, 668,
734, 924, 989, 1078, 1144, 1145
\@xdy@requiredstyles
. . 64, 65, 184, 417, 483, 585,
667, 735, 839, 923, 989, 1077, 1145
\@xdy@sortrules
. . 64, 64, 188, 416, 486, 588,
667, 737, 841, 922, 992, 1077, 1148
\@xdy@style 184, 483,
585, 586, 735, 839, 989, 990, 1145
\@xdy@useralphabets
. . 61, 61, 185, 414, 483, 586,
664, 735, 839, 920, 990, 1075, 1145
\@xdy@userlocationdefs
. 62, 63, 186, 415, 416,
484, 584, 587, 665, 666, 736,
838, 840, 921, 922, 990, 1076, 1146
\@xdy@userlocationnames
. . . 63, 63, 64, 415, 416, 585,
665, 666, 838, 921, 922, 1076, 1077
\@xfor@nextelement
. 215, 502, 754, 1010, 1165
\@xp@gl@getcounterprefix . 209,
212, 213, 1007, 1008, 1163, 1164
\@xp@glshypernumber
. 247, 247, 1025, 1180
\\ 104, 105,
130, 136, 182, 189, 247, 248,
440, 456, 482, 486, 487, 519,
588, 589, 591, 592, 598, 599,
691, 707, 712, 713, 734, 738,
771, 841, 842, 844, 845, 851,
852, 947, 961, 966, 988, 992,
993, 1025, 1026, 1101, 1102,
1116, 1121, 1144, 1148, 1149, 1181
\^ 26, 36,
396, 646, 652, 900, 907, 1055, 1062
_ 221–226, 235, 236
\^ 26, 396, 646, 900, 1055
\| 131, 133,
136, 192, 457, 459, 489, 708,
710, 712, 740, 962–964, 966,
995, 1116, 1118, 1119, 1121, 1151
\~ 26, 397, 646, 900, 1055

A	
<code>\A</code>	244, 1023, 1179
<code>\a</code>	26, 397, 646, 901, 1055
<code>\AA</code>	27, 397, 646, 901, 1055
<code>\aa</code>	27, 397, 646, 901, 1055
<code>\Ac</code>	269, 532, 784, 1039, 1194
<code>\ac</code>	269, 532, 784, 1039, 1194
<code>access (key)</code>	347
<code>\accsuppglossaryentryfield</code>	372, 549, 570, 802, 823
<code>\accsuppglossarysubentryfield</code>	372, 549, 570, 802, 823
<code>\Acf</code>	269, 532, 784, 1039, 1194
<code>\acf</code>	269, 532, 784, 1039, 1194
<code>\Acfp</code>	269, 532, 784, 1039, 1194
<code>\acfp</code>	269, 532, 784, 1039, 1194
<code>\Acl</code>	269, 532, 784, 1039, 1194
<code>\acl</code>	269, 532, 784, 1039, 1194
<code>\Aclp</code>	269, 532, 784, 1039, 1194
<code>\aclp</code>	269, 532, 784, 1039, 1194
<code>\Acp</code>	270, 532, 784, 1039, 1194
<code>\acp</code>	269, 532, 784, 1039, 1194
<code>\acrfootnote</code>	270, 533, 534, 539, 786, 792, 1039, 1195
<code>\ACRfull</code> ...	254, 521, 773, 1028, 1183
<code>\ACRfull</code>	253, 269, 521, 532, 772, 784, 1027, 1028, 1039, 1183, 1194
<code>\acrfull</code>	253, 254, 256, 265, 266, 377, 379, 521, 523, 529, 530, 574, 576, 773, 774, 781, 782, 827, 829, 1027–1029, 1035, 1037, 1183–1185, 1191, 1193
<code>\ACRfullfmt</code>	252, 254, 256, 264, 266, 377, 379, 521, 522, 529, 530, 574, 576, 773, 774, 781, 782, 827, 829, 1027–1029, 1035, 1037, 1183, 1185, 1191, 1193
<code>\acrfullfmt</code>	252, 253, 256, 264, 266, 377, 379, 520, 522, 529, 530, 574, 575, 772, 774, 780, 782, 827, 829, 1027–1029, 1035, 1037, 1183, 1185, 1191, 1192
<code>\acrfullformat</code>	178, 252, 252, 480, 520, 521, 532, 545, 732, 772, 784, 798, 986, 1027, 1142, 1183
<code>\ACRfullpl</code> ..	255, 521, 773, 1028, 1184
<code>\acrfullpl</code> ...	254, 269, 521, 532, 773, 784, 1028, 1039, 1184, 1194
<code>\acrfullpl</code>	254, 254, 269, 521, 532, 773, 784, 1028, 1039, 1184, 1194
<code>\ACRfullplfmt</code>	253, 255, 256, 265, 266, 378, 380, 522, 523, 529, 530, 574, 576, 773–775, 781, 782, 827, 829, 1027–1029, 1036, 1037, 1183–1185, 1191, 1193
<code>\AcRfullplfmt</code>	253, 254, 256, 265, 266, 378, 380, 521, 523, 529, 530, 574, 576, 773, 775, 781, 782, 827, 829, 1027–1029, 1036, 1037, 1183–1185, 1191, 1193
<code>\acrfullplfmt</code>	253, 254, 256, 265, 266, 377, 380, 521, 523, 529, 530, 574, 576, 773, 775, 781, 782, 827, 829, 1027–1029, 1035, 1037, 1183–1185, 1191, 1193
<code>\acrlinkfootnote</code>	270, 270, 533, 786, 1039, 1195
<code>\acrlinkfullformat</code>	252, 252, 253, 520–522, 772–774, 1027, 1183
<code>\ACRlong</code>	166, 475, 726, 980, 1136
<code>\Acrlong</code> ..	166, 166, 269, 474, 532, 726, 784, 980, 1039, 1135, 1194
<code>\acrlong</code>	165, 269, 474, 532, 725, 784, 979, 1039, 1135, 1194
<code>\ACRlongpl</code> ..	168, 476, 727, 981, 1137
<code>\Acrlongpl</code> ..	168, 168, 269, 475, 532, 727, 784, 981, 1039, 1137, 1194
<code>\acrlongpl</code> ...	167, 269, 475, 532, 727, 784, 981, 1039, 1136, 1194
<code>\acrnameformat</code>	255, 522, 537, 580, 774, 790, 833, 1028, 1184
<code>\acrnoLinkfootnote</code>	271, 533, 786, 1039, 1195
<code>acronym (option)</code>	20
<code>\acronymentry</code>	256, 257, 259–263, 265–268, 373–376, 378–382, 522–532, 570–577, 774–784, 824–826, 828–831, 1029–1034, 1036–1038, 1185, 1187–1190, 1192–1194
<code>\acronymfont</code>	120, 161–165, 172, 178, 255, 255, 257, 259–268, 361, 362, 364–366, 373–382, 450, 451, 472–474, 477, 480, 522–532, 534, 535, 537–543, 562–565, 570–577, 579–581, 701, 702, 723–725, 729, 732, 774–784, 786,

788, 789, 791–794, 796, 815–819, 824–832, 834, 955, 956, 977–979, 983, 986, 1028, 1030–1038, 1110, 1111, 1133–1135, 1139, 1142, 1184, 1185, 1187–1194

`\acronymlists` (option) [22](#)

`\acronymname`
. . . [20](#), [47](#), [50](#), [392](#), [393](#), [407](#),
[408](#), [642](#), [643](#), [657](#), [658](#), [897](#),
[912](#), [914](#), [1051](#), [1052](#), [1067](#), [1069](#)

`\acronyms` (option) [20](#)

`\acronymsort` [256](#), [257](#), [259–263](#),
[265–268](#), [373–376](#), [378](#), [379](#),
[381](#), [382](#), [522–532](#), [570–572](#),
[575](#), [577](#), [774–784](#), [824–826](#),
[828–831](#), [1029–1034](#), [1036–](#)
[1038](#), [1185](#), [1187–1190](#), [1192](#), [1194](#)

`\acronymtype` [19](#), [20](#), [256](#), [257](#),
[392](#), [393](#), [522](#), [523](#), [532–546](#),
[578–581](#), [642](#), [643](#), [774](#), [775](#),
[784–799](#), [831–834](#), [896](#), [897](#),
[1029](#), [1030](#), [1051](#), [1052](#), [1184](#), [1185](#)

`\acrpluralsuffix` [251](#), [256](#),
[259–261](#), [265–267](#), [373](#), [374](#),
[378–381](#), [520](#), [522](#), [524–526](#),
[529–536](#), [538–540](#), [542](#), [543](#),
[545](#), [570](#), [571](#), [575](#), [577–582](#),
[772](#), [774](#), [776–778](#), [781–784](#),
[786–788](#), [790–793](#), [795](#), [796](#),
[798](#), [799](#), [824](#), [828–835](#), [1027](#),
[1029](#), [1031–1033](#), [1036–1038](#),
[1182](#), [1185](#), [1187–1189](#), [1192](#), [1193](#)

`\ACRshort` [162](#), [472](#), [724](#), [978](#), [1133](#)

`\Acrshort` [162](#), [162](#), [268](#), [472](#), [532](#),
[724](#), [784](#), [977](#), [1038](#), [1133](#), [1194](#)

`\acrshort` [161](#), [268](#), [472](#), [532](#),
[723](#), [784](#), [977](#), [1038](#), [1133](#), [1194](#)

`\ACRshortpl` [164](#), [473](#), [725](#), [979](#), [1135](#)

`\Acrshortpl` [164](#), [164](#), [268](#), [473](#), [532](#),
[725](#), [784](#), [979](#), [1038](#), [1134](#), [1194](#)

`\acrshortpl` [163](#), [268](#), [473](#), [532](#),
[724](#), [784](#), [978](#), [1038](#), [1134](#), [1194](#)

acrstyles:

`dua` [263](#), [376](#)

`dua-desc` [265](#), [378](#)

`footnote` [265](#), [379](#)

`footnote-desc` [267](#), [381](#)

`footnote-sc` [267](#), [380](#)

`footnote-sc-desc` [268](#), [381](#)

`footnote-sm` [267](#), [381](#)

`footnote-sm-desc` [268](#), [381](#)

`long-sc-short` [260](#)

`long-sc-short-desc` [262](#), [374](#)

`long-short` [258](#), [373](#)

`long-short-desc` [261](#), [374](#)

`long-sm-short` [261](#)

`long-sm-short-desc` [262](#), [375](#)

`long-sp-short` [259](#)

`long-sp-short-desc` [262](#)

`sc-short-long` [261](#)

`sc-short-long-desc` [263](#), [375](#)

`short-long` [260](#), [373](#)

`short-long-desc` [262](#), [375](#)

`sm-short-long` [261](#)

`sm-short-long-desc` [263](#), [376](#)

`\Acs` [268](#), [532](#), [784](#), [1038](#), [1194](#)

`\acs` [268](#), [532](#), [784](#), [1038](#), [1194](#)

`\Acsp` [268](#), [532](#), [784](#), [1038](#), [1194](#)

`\acsp` [268](#), [532](#), [784](#), [1038](#), [1194](#)

`\add@accent@` [26](#), [396](#), [646](#), [900](#), [1055](#)

`\addcontentsline`
. [57](#), [412](#), [662](#), [918](#), [1072](#)

`\addglossarytocaptions` [48](#),
[49](#), [407](#), [657](#), [658](#), [913](#), [1067](#), [1068](#)

`\addtolength` [335](#), [596](#), [634](#), [849](#), [887](#)

`\advance` [17](#), [18](#), [100](#), [107](#), [128](#), [301](#),
[391](#), [392](#), [437](#), [455](#), [614](#), [641](#),
[688](#), [706](#), [868](#), [895](#), [896](#), [944](#),
[948](#), [960](#), [1050](#), [1099](#), [1103](#), [1114](#)

`\AE` [26](#), [397](#), [646](#), [901](#), [1055](#)

`\ae` [26](#), [397](#), [646](#), [901](#), [1055](#)

`altlist` (style) [289](#)

`altlistgroup` (style) [289](#)

`altlisthypergroup` (style) [289](#)

`altlong4col` (style) [296](#)

`altlong4col-booktabs` (style) [299](#)

`altlong4colborder` (style) [296](#)

`altlong4colheader` (style) [296](#)

`altlong4colheaderborder` (style) [297](#)

`altlongragged4col` (style) [305](#)

`altlongragged4col-booktabs`
(style) [300](#)

`altlongragged4colborder` (style) [307](#)

`altlongragged4colheader` (style) [306](#)

`altlongragged4colheaderborder`
(style) [307](#)

`\altnewglossary` [76](#), [423](#), [673](#), [929](#), [1084](#)

`altsuper4col` (style) [319](#)

`altsuper4colborder` (style) [320](#)

`altsuper4colheader` (style) [319](#)

altsuper4colheaderborder (style)	320	907, 1005, 1007, 1025, 1044,
altsuperragged4col (style)	325	1045, 1062, 1161, 1163, 1168, 1181
altsuperragged4colborder (style)	326	
altsuperragged4colheader (style)	326	
altsuperragged4colheaderborder (style)	326	
alttree (style)	333	
alttreegroup (style)	336	
alttreehypergroup (style)	336	
\andname	48 , 215 , 407 , 502 , 657 , 754 , 913 , 1010 , 1067 , 1166	
\AnyTrackedLanguages	49 , 337 , 338 , 408 , 583 , 600 , 658 , 836 , 853 , 914 , 1068	
\appto	22 , 27 , 82 , 91 , 92 , 208 , 339 , 349 , 394 , 397 , 426 , 431 , 551 , 600 , 601 , 644 , 647 , 676 , 682 , 804 , 853 , 854 , 898 , 901 , 932 , 938 , 1007 , 1053 , 1056 , 1087 , 1092 , 1093 , 1162	
\AtBeginDocument	20 , 66 , 88 , 107 , 181 , 201 , 393 , 417 , 429 , 441 , 481 , 495 , 642 , 668 , 680 , 692 , 733 , 747 , 897 , 923 , 936 , 948 , 988 , 1002 , 1051 , 1078 , 1090 , 1103 , 1143 , 1157	
\AtEndDocument	35 , 37 , 88 , 113 , 200 , 204 , 220 , 283 , 402 , 403 , 429 , 444 , 494 , 497 , 505 , 605 , 652 , 653 , 680 , 695 , 746 , 749 , 757 , 858 , 907 , 908 , 936 , 952 , 1001 , 1004 , 1012 , 1042 , 1061 , 1063 , 1090 , 1107 , 1157 , 1160 , 1168	
automake (option)	37	
automakegloss (option)	38	
automakeglosslite (option)	38	
B		
\b	26 , 397 , 646 , 901 , 1055	
\begin	187 , 232 , 288 , 291–297 , 299 , 300 , 302– 327 , 485 , 509 , 587 , 608–612 , 614–629 , 736 , 761 , 840 , 861 , 863–883 , 991 , 1017 , 1147 , 1173	
\BeginAccSupp	346 , 549 , 802	
\begingroup	6 , 8 , 36 , 206 , 211 , 219 , 248 , 385 , 387 , 498 , 500 , 519 , 635 , 636 , 652 , 750 , 752 , 771 , 889 , 891 ,	
\bfseries	292 , 294–296 , 298 , 299 , 303– 305 , 307 , 315 , 317–320 , 322– 327 , 610–613 , 616–618 , 623– 629 , 863–867 , 869–872 , 876–883	
\bgroup	26 , 97 , 136 , 179 , 213 , 227 , 248 , 396 , 435 , 480 , 504 , 506 , 646 , 686 , 712 , 732 , 756 , 758 , 900 , 942 , 966 , 987 , 1008 , 1011 , 1013 , 1026 , 1055 , 1097 , 1121 , 1142 , 1164 , 1169 , 1181	
\bool	138 , 1123	
\boolean	545 , 798	
\boolfalse	38 , 39 , 403 , 653 , 909 , 1063 , 1064	
\booltrue	403 , 653	
\bottomrule	298 , 299 , 613 , 866 , 867	
\box	301 , 614 , 615 , 868	
C		
\c	26 , 236 , 397 , 646 , 901 , 1055	
\c@equation	128 , 455 , 706 , 960 , 1114	
\c@glossaryentry	14 , 390 , 639 , 893 , 1048	
\c@glossarysubentry	15 , 390 , 640 , 894 , 1049	
\c@page	207 , 208 , 211 , 499 , 500 , 751 , 752 , 1006–1008 , 1162 , 1163	
\capitalisewords	175 , 730 , 984 , 1140	
\catcode	36 , 88 , 136 , 429 , 652 , 680 , 712 , 907 , 936 , 966 , 1062 , 1090 , 1121	
\cGls	115 , 115 , 445 , 696 , 953 , 1108	
\cgl	114 , 114 , 115 , 445 , 696 , 953 , 1107 , 1108	
\cGlsformat	113 , 115 , 444 , 445 , 695 , 696 , 952 , 953 , 1107 , 1108	
\cglformat	112 , 114 , 444 , 445 , 695 , 696 , 952 , 953 , 1106 , 1108	
\cGlspl	116 , 116 , 445 , 446 , 696 , 953 , 1108	
\cglsp	115 , 115 , 116 , 445 , 696 , 953 , 1108	
\cGlsplformat	113 , 116 , 444 , 446 , 695 , 697 , 952 , 954 , 1107 , 1108	
\cglspformat	113 , 116 , 444 , 445 , 695 , 696 , 952 , 953 , 1107 , 1108	
\changes	179	

<code>\char</code>	244, 517, 769, 1024, 1179	<code>\csname</code> .	15–18, 26, 42, 47, 49, 50,
<code>\cleardoublepage</code>	56, 57, 411, 661, 662, 917, 1072		56, 59, 62, 65, 67, 70, 76, 77,
<code>\clearpage</code>	56, 57, 411, 661, 662, 917, 1072		84, 85, 91–93, 95, 96, 100–107,
<code>\closeout</code> 36, 88, 188, 190, 194, 205,			109, 110, 127–129, 141–145,
402, 429, 486, 487, 490, 498,			162–169, 179, 181, 185, 186,
652, 680, 737, 739, 742, 750,			191–194, 200, 204–206, 208,
907, 936, 992, 993, 997, 1005,			212–214, 218, 220, 228, 246,
1062, 1090, 1148, 1149, 1153, 1160			250, 274–279, 333, 335, 339,
<code>\compatglossarystyle</code>			350, 351, 354, 358, 364–366,
.	589–599, 843–853		383, 384, 390–392, 396, 405,
<code>compatible-2.07</code> (option)	38		407, 408, 411, 413, 415, 417–
<code>compatible-3.07</code> (option)	38		419, 422, 423, 427, 431–434,
<code>\compatibleglossentry</code>	241,		437–442, 452, 454–456, 463–
346, 515, 516, 549, 767, 768, 802			466, 472–476, 481, 483–485,
<code>\compatiblesubglossentry</code>			488–490, 495, 497–499, 501,
.	243, 346, 516, 549, 768, 802		502, 504–506, 512, 515, 516,
<code>\copy</code>	301, 614, 615, 868		518, 520, 546–549, 551–554,
<code>\count@</code> 233, 510, 762, 1017, 1018, 1173			557, 564, 565, 582–585, 596,
<code>counter</code> (key)	80		601, 605, 606, 633, 634, 640–
<code>counter</code> (option)	22		642, 646, 655, 657, 658, 661,
counters:			663, 665, 667, 668, 670, 672,
<code>glossaryentry</code>	14		673, 677, 678, 681–685, 688–
<code>glossarysubentry</code>	15		693, 703, 705–707, 715–717,
<code>counterwithin</code> (option)	14		723–728, 732, 733, 735, 736,
<code>\cs</code>	43–45, 221–226, 236, 281, 282		740–742, 746, 749–751, 753–
<code>\csdef</code>	24, 91–93, 103, 104,		758, 764, 767–770, 772, 785,
111, 112, 226, 228, 246, 258,			799–802, 805–807, 810, 817–
395, 431–433, 439, 440, 443,			819, 835–838, 849, 854, 858,
444, 506, 518, 523, 524, 589,			859, 886, 887, 894–896, 900,
645, 682, 683, 690, 691, 694,			911–914, 917, 919, 921, 923–
757, 758, 770, 775, 843, 899,			925, 928, 929, 933, 934, 937–
938, 939, 946, 947, 951, 1013,			941, 944–948, 950, 959, 960,
1014, 1025, 1030, 1054, 1092–			969–971, 977–982, 987, 988,
1094, 1101, 1106, 1169, 1180, 1186			990, 991, 995–997, 1002, 1004–
<code>\csedef</code>	114, 208, 444, 499,		1006, 1008, 1009, 1011, 1012,
695, 751, 952, 1006, 1107, 1162			1014, 1024, 1026, 1040–1042,
<code>\csgdef</code>	19, 54, 74, 78,		1049–1051, 1055, 1066–1069,
112, 113, 216, 235, 236, 410,			1071, 1072, 1074–1076, 1078–
422, 423, 443, 444, 503, 512,			1080, 1083, 1084, 1088, 1092–
660, 672, 673, 694, 695, 755,			1096, 1098–1103, 1105, 1114,
764, 896, 916, 928, 929, 951,			1115, 1124–1126, 1133–1137,
952, 1010, 1019, 1051, 1070,			1143, 1145–1147, 1150–1153,
1082, 1084, 1106, 1107, 1166, 1175			1157, 1160–1162, 1164, 1165,
<code>\cslet</code>	81, 82, 97, 104,		1167–1169, 1180, 1182, 1195–1197
231, 426, 435, 440, 509, 676,		<code>\csshow</code>	278, 548, 801, 1041, 1197
686, 691, 761, 931, 932, 942,		<code>\csuse</code>	50, 53, 54, 74, 82,
947, 1016, 1086, 1097, 1101, 1172			84, 85, 92, 93, 169, 195, 196,
			227, 231, 233, 235, 237, 238,
			258, 339, 408, 410, 422, 427,
			428, 432, 433, 452, 453, 491,

492, 506, 508–510, 512, 513, 518, 524, 590–599, 601, 658, 660, 672, 677, 678, 683, 703, 743, 744, 758, 760–762, 764, 765, 770, 776, 844–848, 850– 854, 914, 915, 928, 931, 933, 934, 939, 982, 998, 999, 1014, 1015, 1017–1020, 1030, 1069, 1070, 1083, 1086, 1088, 1089, 1093, 1094, 1137, 1153–1155, 1169, 1171–1173, 1175, 1176, 1186	<code>\DeclareCurrentRelease</code>
<code>\csxdef</code> 102, 113, 438, 444, 689, 695, 785, 945, 952, 1100, 1107 4, 280, 284, 286, 290, 297, 302, 308, 313, 321, 327, 337, 338, 345, 384, 385, 635, 888
<code>\currentglossary</code> 14, 15, 54, 219, 280, 282, 283, 390, 410, 504, 639, 640, 660, 756, 894, 915, 1012, 1042, 1048, 1049, 1070, 1167	<code>\DeclareListParser</code>
<code>\currentglssubentry</code> 199, 494, 745, 1000, 1156
. 15, 240, 390, 514, 640, 766, 894, 1021, 1049, 1177	<code>\DeclareOption</code> . 11, 338, 345, 388, 548, 600, 638, 801, 853, 892, 1047
<code>\CurrentOption</code>	<code>\DeclareOptionX</code> 11, 388, 638, 892, 1047
. 42, 338, 345, 405, 548, 600, 655, 801, 853, 911, 1065, 1066	<code>\DeclareRelease</code>
<code>\CurrentTrackedLanguage</code> 4, 280, 284, 286, 290, 297, 302, 308, 313, 321, 327, 337, 338, 345, 384, 385, 635, 888
. 50, 337, 338, 408, 583, 600, 658, 837, 853, 914, 1068	<code>\DeclareRobustCommand</code>
<code>\CurrentTrackedTag</code>	50, 408, 541, 659, 793, 914, 1069
. 49, 337, 338, 408, 583, 600, 658, 836, 853, 914, 1068	<code>\def</code> 11, 12, 14–21, 25–27, 33–35, 37, 38, 40, 45–47, 50, 51, 54, 58, 61–66, 70, 75–83, 86, 95, 97, 99–104, 107, 108, 112–117, 123, 127–137, 140– 169, 179–181, 184, 188, 190– 197, 200–202, 204, 208, 209, 211–214, 218, 226, 231–238, 244–250, 253–256, 283–286, 312, 313, 333–336, 338, 339, 341–345, 347–350, 364–371, 388–393, 396, 397, 401, 403– 410, 412, 414–420, 422–426, 428, 434–441, 444–446, 452– 476, 481, 483, 486–493, 495– 497, 499–502, 504, 505, 509– 513, 517–522, 533–538, 540– 542, 544, 546, 549–551, 564– 569, 578–582, 585, 588, 590, 595, 596, 600, 602–607, 621, 622, 633, 634, 638–643, 645– 647, 651–654, 656–660, 662, 664–670, 672–676, 678, 684, 686–692, 694–697, 703–728, 732, 733, 735, 737, 739–744, 747–749, 751–757, 761–765, 769–774, 785, 787–795, 797, 799, 802–805, 817–822, 832– 835, 838, 839, 841, 843, 849, 853–857, 859, 860, 874, 875, 886–888, 892–898, 900, 901, 905, 907–912, 914–916, 918, 920–925, 928–932, 934, 941– 949, 951–954, 957, 959–982, 987–989, 992, 994–1004, 1006–
<code>\CustomAcronymFields</code>	
. 273, 545, 798, 799	
<code>\CustomNewAcronymDef</code>	
. 273, 545, 546, 798, 799	
D	
<code>\d</code> 26, 396, 646, 900, 1055	
<code>\datatool</code> 221, 224, 225	
<code>\datatoolasciistart</code> 223	
<code>\datatoolctrlboundary</code> 223–225	
<code>\day</code> 184, 188, 483, 486, 585, 588, 735, 738, 839, 841, 989, 992, 1145, 1148	
<code>debug</code> (option) 7	
<code>\DeclareAcronymList</code>	
20, 21, 22, 256, 257, 392–394, 522, 523, 533, 535, 536, 538, 540, 542, 544, 546, 642–644, 774, 775, 785, 787, 789, 791, 793, 795, 797, 799, 897, 898, 1029, 1030, 1051–1053, 1184, 1185	

1011, 1013, 1016, 1017, 1019–
1021, 1023–1029, 1042, 1047–
1052, 1054–1056, 1060, 1061,
1063, 1064, 1066, 1067, 1069,
1070, 1073, 1075–1078, 1080,
1083–1087, 1089, 1095, 1097–
1101, 1103, 1104, 1106–1109,
1112, 1114–1122, 1124–1137,
1142, 1143, 1145, 1148–1155,
1157, 1158, 1160, 1162–1165,
1167, 1168, 1172–1176, 1179–1184
\def@glsexdycheckbackslash
137, 461, 462, 713, 714, 967, 1122
\DefaultNewAcronymDef 270,
382, 532, 533, 578, 784, 786, 831
\defgldisplay 122, 452, 703
\defgldisplayfirst . . . 122, 452, 703
\defglentryfmt
74, 76, 78, 257, 270, 422, 423,
452, 523, 532, 533, 535, 537,
539, 541, 543, 545, 672–674,
703, 775, 784, 786, 788, 789,
791, 794, 796, 798, 928, 929,
1030, 1039, 1082–1084, 1186, 1195
\define@boolkey 10, 12–
15, 20, 27, 28, 31–33, 38, 123,
124, 238, 388–390, 392, 397,
398, 400, 401, 403, 453, 513,
637–640, 642, 647, 650, 651,
653, 704, 765, 892–894, 897,
901, 902, 904, 905, 908, 909,
957, 1020, 1046–1049, 1051,
1056, 1059, 1060, 1063, 1112, 1176
\define@choicekey . . 7, 9, 10, 15,
28, 30, 33, 37, 40, 41, 81, 237,
238, 386–388, 390, 398, 399,
401, 402, 404, 425, 512, 513,
635, 637, 638, 640, 648–650,
653, 654, 675, 764, 765, 889,
891, 892, 894, 902, 903, 905,
907, 909, 910, 931, 1020, 1044,
1046, 1049, 1057, 1058, 1060,
1062, 1064, 1065, 1086, 1175, 1176
\define@key 11, 14, 22, 23,
29, 31, 33, 34, 78–83, 91, 92,
123, 180, 236, 238, 338, 347–
349, 388, 390, 394, 398, 401,
424–426, 431, 453, 481, 512,
513, 549, 550, 600, 638, 640,
644, 648, 651, 674–676, 682,
704, 733, 764, 765, 803, 804,
853, 892, 894, 898, 902, 904,
905, 930–932, 938, 957, 988,
1019–1021, 1047, 1048, 1053,
1057, 1059, 1060, 1084–1087,
1092, 1093, 1112, 1143, 1175, 1176
\DefineAcronymSynonyms
. 45, 268, 270, 406, 532, 656,
784, 911, 1038, 1039, 1066, 1194
\delimN 53, 187, 199,
235, 248, 410, 485, 494, 511,
519, 587, 660, 737, 745, 763,
771, 841, 915, 991, 1001, 1019,
1025, 1070, 1147, 1156, 1174, 1181
\delimR 53, 187, 247, 248, 410, 485,
519, 587, 660, 737, 771, 841,
915, 991, 1025, 1070, 1147, 1181
description (key) 78
description (option) 32
descriptionaccess (key) 348
\DescriptionDUANewAcronymDef . .
. 271, 535, 537, 788, 789
\DescriptionFootnoteNewAcronymDef
. 271,
382, 534, 535, 579, 786, 787, 832
\descriptionname
. . 47, 50, 292, 294–296, 298,
299, 303–305, 307, 315, 317–
320, 322–327, 407, 408, 610–
613, 616–618, 623–625, 627–
629, 657, 658, 863–867, 869–
872, 876–883, 912, 914, 1067, 1069
\DescriptionNewAcronymDef . 271,
382, 537, 538, 579, 790, 791, 833
descriptionplural (key) 79
descriptionpluralaccess (key) . . 348
\DH 27, 397, 647, 901, 1056
\dh 27, 397, 647, 901, 1056
\dimen@ . . 260, 301, 333, 525, 614,
632, 633, 777, 868, 886, 1032, 1187
\directlua 43, 911, 1066
\disable@keys . 45, 405, 655, 911, 1066
disablemakegloss (option) 34
\do 31, 42, 46, 48, 58, 60,
67, 89, 90, 130, 179, 184–186,
197–199, 207, 215, 226, 227,
257, 400, 405, 406, 412, 413,
418, 429, 430, 456, 481, 483,
484, 493, 494, 499, 502, 506,
523, 533, 535, 537, 538, 540,

543, 544, 546, 585, 605, 606, 650, 655, 657, 663, 668, 680, 681, 707, 732, 735, 736, 745, 746, 751, 754, 757, 758, 775, 785, 788, 789, 791, 793, 796, 797, 799, 839, 858, 859, 904, 911–913, 918, 919, 924, 936, 937, 961, 987, 989–991, 999– 1001, 1006, 1010, 1013, 1030, 1042, 1059, 1065, 1067, 1068, 1073, 1074, 1078, 1079, 1091, 1092, 1116, 1143, 1145, 1146, 1155, 1156, 1161, 1165, 1169, 1186	\DTLsortletterhandler 224
	\DTLsortwordcasehandler 224, 225
	\DTLsortwordhandler 224
	\DTLsubstituteall 130, 457, 707, 961, 1116
	\dtlwordindexcompare 228, 507, 759, 1014, 1170
	dua (acrstyle) 263, 376
	dua (option) 32
	dua-desc (acrstyle) 265, 378
	\DUANewAcronymDef 272, 543, 544, 796, 797
	E
	\eappto 423, 424, 440, 499
	\edef 46, 49, 58–63, 65, 76, 84, 85, 88, 90, 128–137, 182, 183, 188, 190– 192, 199, 204, 220, 228, 229, 233, 244, 248–250, 392, 393, 406, 408, 412–419, 422–424, 427–430, 433–436, 440, 452, 454–462, 480–482, 486–489, 491, 492, 494, 497, 500, 501, 503, 505–507, 510, 514, 517, 519, 520, 532, 534, 535, 537, 539, 541, 543, 578–581, 584– 586, 588, 605, 657, 658, 662– 667, 672, 673, 677, 678, 680, 681, 703, 706–714, 733, 734, 737, 739–741, 743–745, 749, 752, 753, 757, 759, 762, 769, 771, 772, 784, 786, 788, 790, 792, 794, 796, 831–834, 837– 839, 841, 912, 913, 918–923, 928, 933, 934, 936, 937, 959– 967, 988, 992, 994–1000, 1004, 1012, 1014, 1015, 1018, 1024, 1026, 1067, 1068, 1072–1077, 1083, 1088, 1090, 1092, 1114– 1122, 1144, 1148–1156, 1160, 1168, 1170, 1173, 1179, 1181, 1182
\do@gl@storeentry 16–19, 103, 390–392, 439, 640–642, 690, 894–896, 946, 1049–1051, 1101	\egroup 26, 98, 180, 213, 227, 249, 396, 435, 481, 504, 506, 646, 686, 733, 756, 758, 900, 942, 987, 1009, 1012, 1013, 1026, 1055, 1097, 1143, 1164, 1169, 1181
\do@gl@link@checkfirsthyper 125, 127, 140–146, 161– 169, 364–366, 454, 463–466, 472–476, 564, 565, 704, 705, 715–718, 723–727, 817, 818, 958, 959, 968–971, 977–982, 1113, 1114, 1124–1127, 1133–1137	\else 8, 13–15, 18–21, 23, 25, 27, 28, 35, 37, 38, 42, 45, 46, 51, 52, 55, 57–67, 74, 81, 84, 100,
\do@gl@x@dycheckbackslash 130, 456, 707, 961, 1116	
\do@gl@disablehyperinlist 126, 127, 454, 705, 959, 1113, 1114	
\do@gl@shaschildren 70, 419, 420, 669, 670, 925, 1080	
\do@write@gl@labelsname 35, 36, 36, 652, 907, 1061, 1062	
\doifglossarynoexistsordo 70, 75, 419, 422, 669, 672, 925, 928, 1080, 1083	
\dtl@ifsingle 517, 769	
\dtl@insertinto 228, 507, 759, 1014, 1170	
\dtl@sortresult 228, 229, 507, 759, 1014, 1015, 1170	
\DTLassignlettergroup 225	
\dtlcompare 229, 507, 759, 1015, 1170	
\dtlicompare 229, 507, 759, 1015, 1170	
\DTLifinlist 78, 126, 424, 454, 674, 705, 930, 959, 1084, 1113	
\DTLifint 244, 517, 769, 1024, 1179	
\dtlletterindexcompare 229, 507, 759, 1014, 1170	
\dtlnonlettergroup 225	
\dtlnumbergroup 225	
\DTLsortedletter 225	

101, 104–107, 112, 113, 124,
126–137, 172, 183, 184, 187–
197, 204, 206–209, 211–215,
218, 233, 238, 240, 245, 247–
249, 260, 272, 274, 288, 292,
293, 295, 298, 299, 301, 303,
304, 306, 315, 316, 318, 322,
324, 326, 329, 331, 332, 334,
335, 355, 386, 387, 389, 390,
392–395, 397, 398, 402–406,
409–418, 425, 427, 437, 438,
440, 441, 444, 454–466, 477,
482, 483, 485–492, 497–502,
504, 510, 513–515, 517, 519,
525, 535, 536, 538, 540, 541,
543–546, 554, 585–590, 594–
596, 605, 608, 610, 611, 613–
617, 622–624, 626–628, 630–
634, 636, 639–645, 647, 648,
652–657, 659–668, 675, 677,
688, 689, 691, 692, 695, 705–
717, 728, 729, 734, 735, 737,
739–744, 749–754, 756, 762,
765–767, 769, 771, 777, 787–
789, 791, 793–796, 798, 799,
808, 838, 839, 841–843, 847–
849, 858, 861, 863–871, 876–
878, 880–882, 884–888, 890,
893–899, 901, 902, 907, 908,
911, 912, 915–924, 928, 931,
933, 944, 945, 947, 948, 952,
957–967, 983, 989, 990, 992–
994, 996–1000, 1005–1010,
1012, 1017, 1021, 1022, 1024–
1026, 1032, 1039, 1042, 1045,
1048–1054, 1056, 1057, 1062,
1063, 1065–1067, 1069–1079,
1082, 1086, 1088, 1098, 1099,
1101–1103, 1106, 1107, 1112–
1122, 1139, 1144, 1145, 1147–
1155, 1160–1165, 1167, 1173,
1176–1178, 1180–1182, 1187, 1195
\emph 215, 249, 502, 520,
754, 772, 1010, 1026, 1165, 1182
\empty 248,
346, 519, 548, 771, 802, 1026, 1181
\encodingdefault
. 26, 396, 646, 900, 1055
\end 187, 232,
288, 291–297, 299, 300, 302–
327, 485, 510, 587, 608–612,
614–629, 736, 761, 840, 861,
863–883, 991, 1017, 1147, 1173
\end@doifinlist 58, 412, 662, 918, 1073
\end@getprefix
. 213, 501, 753, 1009, 1164
\end@glis@islistofacronyms
21, 393, 394, 643, 897, 898, 1052
\EndAccSupp 346, 549, 802
\endcsname 15–18, 26, 42, 47, 49, 50,
56, 59, 62, 65, 67, 70, 76, 77,
84, 85, 91–93, 95, 96, 100–107,
109, 110, 127–129, 141–145,
162–169, 179, 181, 185, 186,
191–194, 200, 204–206, 208,
212–214, 218, 220, 228, 246,
250, 274–279, 333, 335, 339,
350, 351, 354, 358, 364–366,
383, 384, 390–392, 396, 405,
407, 408, 411, 413, 415, 417–
419, 422, 423, 427, 431–434,
437–442, 452, 454–456, 463–
466, 472–476, 481, 483–485,
488–490, 495, 497–499, 501,
502, 504–506, 512, 515, 516,
518, 520, 546–549, 551–554,
557, 564, 565, 582–585, 596,
601, 605, 606, 633, 634, 640–
642, 646, 655, 657, 658, 661,
663, 665, 667, 668, 670, 672,
673, 677, 678, 681–685, 688–
693, 703, 705–707, 715–717,
723–728, 732, 733, 735, 736,
740–742, 746, 749–751, 753–
758, 764, 767–770, 772, 785,
799–802, 805–807, 810, 817–
819, 835–838, 849, 854, 858,
859, 886, 887, 894–896, 900,
911–914, 917, 919, 921, 923–
925, 928, 929, 933, 934, 937–
941, 944–948, 950, 959, 960,
969–971, 977–982, 987, 988,
990, 991, 995–997, 1002, 1004–
1006, 1008, 1009, 1011, 1012,
1014, 1024, 1026, 1040–1042,
1049–1051, 1055, 1066–1069,
1071, 1072, 1074–1076, 1078–
1080, 1083, 1084, 1088, 1092–
1096, 1098–1103, 1105, 1114,
1115, 1124–1126, 1133–1137,

1143, 1145–1147, 1150–1153,
 1157, 1160–1162, 1164, 1165,
 1167–1169, 1180, 1182, 1195–1197
`\endfoot` 292–296,
 298, 299, 303–305, 307, 610–
 613, 616–618, 863–867, 869–872
`\endgroup` 6,
 8, 36, 206, 211, 220, 248, 385,
 387, 498, 500, 519, 635, 637,
 652, 750, 752, 771, 889, 891,
 907, 1005, 1008, 1026, 1044,
 1045, 1062, 1161, 1163, 1168, 1181
`\endhead` 292–296,
 298, 299, 303–305, 307, 610–
 613, 616–618, 863–867, 869–872
`\endtheglossary` 6, 385, 635, 889, 1044
`entrycounter` (option) 14
`\entryname` 47, 50, 292, 294–296, 298,
 299, 303–305, 307, 315, 317–
 320, 322–327, 407, 408, 610–
 613, 616–618, 623–625, 627–
 629, 657, 658, 863–867, 869–
 872, 876–883, 912, 914, 1067, 1069
 environments:
 `theglossary` 241
`\epreto` 107, 948, 1103
`\equal` 29, 46, 56, 128, 198, 398,
 406, 411, 455, 493, 517, 605,
 648, 656, 661, 706, 745, 769,
 858, 902, 912, 917, 959, 1000,
 1042, 1057, 1066, 1072, 1114, 1156
`esclocations` (option) 12
`\exp` 44, 221–225, 283
`\expandafter` 8, 16–18, 25, 26, 42, 46,
 49, 59, 61, 62, 64–67, 70, 76–
 78, 84, 85, 89–93, 95, 96, 100–
 110, 122, 126, 128–137, 170–
 172, 175, 180, 182, 183, 186,
 190–192, 194, 204–207, 209,
 211, 212, 215, 218, 220, 226,
 228, 232, 242, 246, 248, 250,
 274–279, 333, 339, 350, 351,
 354, 355, 382–384, 387, 391,
 392, 396, 405–408, 413, 415–
 419, 422–424, 427–434, 437–
 442, 452, 454–462, 477, 481,
 482, 484, 485, 487–490, 497–
 502, 504, 506, 510, 511, 515,
 516, 518–520, 534, 539, 546–
 549, 551–555, 578, 582–585,
 587, 588, 601, 605, 606, 633,
 636, 640–642, 645, 646, 655,
 657, 658, 663, 665, 667, 668,
 670, 672–674, 677, 678, 680–
 685, 688–693, 703, 705–713,
 728–730, 733, 734, 736, 739–
 742, 749–754, 756, 758, 762,
 763, 767, 768, 770–772, 785,
 786, 792, 799–802, 805–808,
 831, 835–838, 840, 841, 854,
 858, 859, 886, 891, 895, 896,
 900, 911–913, 919, 921–925,
 928–930, 933, 934, 936–941,
 944–950, 957, 959–967, 982,
 983, 985, 987, 988, 991, 994–
 997, 1004–1008, 1010, 1012,
 1014, 1017, 1018, 1022, 1024,
 1026, 1040–1043, 1045, 1049–
 1051, 1054, 1055, 1066–1068,
 1073–1080, 1083, 1084, 1088,
 1089, 1091–1096, 1098–1105,
 1111–1122, 1138–1140, 1143,
 1144, 1146, 1147, 1150–1152,
 1160–1164, 1166–1170, 1173,
 1174, 1178, 1180–1182, 1195–1197
`\expandonce`
 84–86, 96, 130, 131, 191–
 193, 208, 228, 229, 256, 427,
 428, 456, 457, 488–490, 499,
 507, 515, 516, 522, 532, 534,
 536, 538, 539, 542, 543, 549,
 677, 678, 685, 707, 708, 740,
 741, 751, 759, 767, 768, 774,
 784–788, 790, 792, 795, 796,
 802, 933, 934, 941, 961, 962,
 995, 996, 1006, 1014, 1015,
 1029, 1088, 1089, 1096, 1116,
 1117, 1150–1152, 1162, 1170, 1185
`\ExplSyntaxOff` 6, 45, 107, 138, 226,
 236, 244, 245, 284, 889, 1023,
 1024, 1043, 1103, 1123, 1179, 1180
`\ExplSyntaxOn` 4, 43, 107, 138,
 221, 235, 244, 245, 281, 888,
 1023, 1024, 1043, 1103, 1123, 1179

F
`\fi` 7, 8, 10, 11, 13–16,
 18–21, 23, 25, 27, 28, 30, 32,
 34, 35, 37, 38, 40–42, 45, 46,
 50–52, 55, 57–67, 74, 77, 81,

84, 96, 100–107, 112, 113, 124,
126–137, 172, 181, 183, 184,
187, 189–195, 197, 199, 204–
209, 211–216, 218, 220, 221,
233, 237–240, 245–249, 260,
270, 272, 274, 288, 292, 293,
295, 298, 299, 301, 303, 305,
306, 315, 316, 318, 322, 324,
326, 329–335, 346, 355, 385–
398, 400, 402–406, 408–418,
423, 425, 427, 437–441, 444,
454–466, 477, 481–483, 485,
487–491, 493, 494, 498–505,
510, 513–515, 517–520, 525,
532, 533, 535, 536, 538–546,
548, 549, 555, 584–590, 594–
596, 605, 608, 610, 611, 613–
617, 622–624, 626–628, 630–
645, 647–649, 652–657, 659–
668, 673, 675, 677, 685, 687–
692, 695, 705–718, 729, 733–
735, 737, 739–746, 749–757,
762, 765–767, 769–771, 777,
784, 785, 787–789, 791, 793–
799, 801, 802, 808, 837–839,
841–843, 847–849, 858, 861,
863–871, 876–878, 880–882,
884–899, 901, 902, 904–912,
914–924, 928, 929, 931, 933,
941, 943–948, 951, 952, 957–
967, 983, 988–990, 992–994,
996–1001, 1005–1013, 1018,
1020–1022, 1024–1026, 1032,
1039, 1040, 1042, 1044–1054,
1056, 1057, 1059, 1060, 1062–
1067, 1069–1079, 1082, 1084,
1086, 1088, 1096, 1098–1100,
1102, 1103, 1106, 1107, 1112–
1122, 1139, 1143–1145, 1147,
1149, 1150, 1152, 1154–1156,
1160–1168, 1173, 1176–1178,
1180–1182, 1187, 1194, 1195

`\findrootlanguage` 65, 417, 667
`first` (key) 79
`firstaccess` (key) 347
`\firstacronymfont` 122, 255,
258–260, 266, 363, 364, 373,
374, 379, 452, 522, 524, 525,
530, 534, 537, 539, 541, 564,
570, 571, 575, 702, 703, 774,
776, 777, 782, 786, 790, 792,
794, 817, 823, 824, 828, 957,
1028, 1031, 1032, 1036, 1037,
1111, 1112, 1184, 1186–1188, 1192
`firstplural` (key) 79
`firstpluralaccess` (key) 347
`\fmtversion` 27, 397, 647, 901, 1056
`\footnote` 266, 270, 271,
379, 530, 533, 575, 782, 786,
828, 1036, 1037, 1039, 1192, 1195
`footnote` (acrstyle) 265, 379
`footnote` (option) 32
`footnote-desc` (acrstyle) 267, 381
`footnote-sc` (acrstyle) 267, 380
`footnote-sc-desc` (acrstyle) 268, 381
`footnote-sm` (acrstyle) 267, 381
`footnote-sm-desc` (acrstyle) 268, 381
`\FootnoteNewAcronymDef` 272,
382, 539, 540, 580, 792, 793, 834
`\footnotesize` 8, 387, 636, 890, 1045
`\forallacronyms` 67, 418, 668, 924, 1079
`\forallglossaries` 66, 67, 204,
217, 333, 418, 497, 503, 632,
668, 749, 755, 886, 924, 1004,
1011, 1078, 1079, 1160, 1166, 1167
`\forallglsentries` 35, 67, 110, 111,
114, 181, 182, 402, 418, 442,
443, 445, 482, 652, 668, 693–
695, 733, 907, 924, 950–952,
988, 1061, 1079, 1105, 1107, 1144
`\ForEachTrackedDialect`
. 49, 337, 338, 408,
583, 600, 658, 836, 853, 914, 1068
`\forglsentries`
. 67, 67, 70, 104, 231, 333,
418, 419, 440, 509, 632, 668,
670, 691, 761, 886, 924, 925,
947, 1016, 1079, 1080, 1101, 1172
`\forallistcsloop`
. 222, 226, 232, 506, 510,
757, 761, 1013, 1017, 1169, 1173
`\forallistloop` 202, 234, 496, 511, 748,
763, 1003, 1019, 1158, 1159, 1174

G

`\gdef` 17, 36, 59, 75, 76, 95, 100, 101,
206, 238, 239, 283, 391, 413,
422, 433, 437, 498, 513, 514,
605, 641, 652, 663, 672, 684,
688, 750, 765, 858, 895, 907,

919, 928, 940, 944, 1005, 1021, 1042, 1050, 1062, 1074, 1083, 1095, 1098, 1099, 1161, 1176, 1177	<code>\glo@name</code> .. 242, 515, 767, 1022, 1178
<code>\Genacrfullformat</code> 121, <u>122</u> , 256– 260, 266, 363, <u>364</u> , 373, 374, 379, 451, 452, 523–525, 530, 563, 564, 570, 571, 575, 702, 774–777, 782, 816, 817, 823, 824, 828, 956, 957, 1029, 1031, 1032, 1036, 1111, 1185–1188, 1192	<code>\glo@parent</code> ... 70, 419, 670, 925, 1080
<code>\genacrfullformat</code> 121, <u>121</u> , 122, 256–260, 266, 363, <u>363</u> , 373, 379, 451, 452, 522–525, 530, 563, 570, 571, 575, 702, 703, 774–777, 782, 816, 817, 823, 824, 828, 956, 957, 1029, 1031, 1032, 1036, 1111, 1185–1187, 1192	<code>\glo@type</code> ... 104, 440, 691, 947, 1101
<code>\GenericAcronymFields</code>	<code>\glo@value</code> ... 89, 429, 680, 936, 1091
.... 256, <u>257</u> , 258–268, 373– 379, 381, 382, 522–532, 570– 572, 574, 575, 577, 774–783, 823–828, 830, 831, 1029–1036, 1038, 1185–1187, 1189–1192, 1194	<code>\global</code> 17, 18, 84, 87, 97, 103, 110, 206, 219, 233, 234, 239, 301, 391, 392, 427, 429, 435, 439, 442, 498, 504, 510, 511, 514, 615, 641, 642, 677, 679, 686, 690, 693, 750, 756, 762, 763, 766, 868, 895, 896, 933, 935, 942, 946, 950, 1005, 1012, 1018, 1021, 1050, 1051, 1088, 1090, 1097, 1101, 1105, 1161, 1168, 1173, 1174, 1177
<code>\Genplacrfullformat</code>	<code>\glolinkprefix</code> 127, <u>128</u> , 180, 241, 455, 481, 515, 706, 733, 767, 959, 988, 1022, 1114, 1143, 1178
121, <u>122</u> , 256, 257, 259, 260, 266, 363, <u>364</u> , 373, 374, 379, 451, 452, 523–525, 530, 563, 564, 570, 571, 575, 702, 703, 775–777, 782, 816, 817, 823, 824, 828, 956, 957, 1029–1032, 1037, 1111, 1112, 1185–1188, 1192	<code>\glosortentrieswarning</code> 23, <u>23</u> , 226, 394, 395, 505, 644, 757, 898, 899, 1013, 1053, 1168
<code>\genplacrfullformat</code>	<code>\glossaries</code>
121, 122, <u>122</u> , 256, 257, 259, 260, 266, 363, <u>364</u> , 373, 374, 379, 451, 452, 523–525, 530, 563, 564, 570, 571, 575, 702, 703, 775–777, 782, 816, 817, 823, 824, 828, 956, 957, 1029–1032, 1037, 1111, 1112, 1185–1188, 1192	<code>\GlossariesWarning</code>
<code>\GetTitleStringDisableCommands</code> 287, 861	7, 9, 23, <u>23</u> , 24, 27–29, 34, 35, 37, 39, 40, 48, 54, 57, 69, 73, 74, 81, 84, 87–89, 114–116, 179, 193, 195, 196, 200–203, 206, 208, 213, 216–218, 272, 346, 386, 388, 394, 395, 397, 398, 401–404, 410, 412, 419, 421, 425, 427, 445, 446, 452, 453, 480, 491, 492, 494–498, 501, 503, 504, 516, 518, 548, 584, 635–637, 644, 645, 647, 648, 651–654, 660, 662, 669, 672, 675, 677, 679, 680, 696, 703, 704, 732, 743, 744, 746–750, 753, 755, 756, 768, 770, 797, 802, 837, 889–891, 898, 899, 901, 902, 906, 907, 909, 913, 916, 918, 925, 927, 928, 931, 933, 935, 936, 953, 987, 997– 999, 1001–1006, 1009, 1011, 1039, 1044–1046, 1053, 1054, 1056, 1057, 1060–1062, 1064, 1068, 1070, 1072, 1079, 1082, 1086, 1087, 1090, 1091, 1108, 1142, 1152–1155, 1157–1159, 1161, 1162, 1165–1167, 1195
<code>\GetTitleStringSetup</code> 287, 861	<code>\GlossariesWarningNoLine</code>
<code>\glo@desc</code> 590, 843	
<code>\glo@do@compare</code> 228, 229, 507, 759, 1014, 1015, 1170	
<code>\glo@grabfirst</code> 226, <u>232</u> , 510, 511, 762, 763, 1017, 1018, 1173, 1174	
<code>\glo@label</code> 70, 104, 419, 440, 670, 691, 925, 947, 1080, 1101	
<code>\glo@list</code> ... 104, 440, 691, 947, 1101	

. 7, 23, 23, 43, 199, 201,
 204, 221, 283, 386, 394, 395,
 493–495, 497, 505, 605, 635,
 636, 644, 746, 747, 749, 757,
 858, 889, 890, 898, 899, 1001,
 1002, 1004, 1005, 1013, 1042,
 1044, 1045, 1053, 1157, 1160, 1168
 \glossary 39, 39, 40, 403,
 404, 585, 653, 654, 838, 909, 1064
 \glossaryentry
 213, 214, 501, 502, 585, 753,
 754, 838, 1008, 1009, 1164, 1165
 glossaryentry (counter) 14
 \glossaryentryfield 243, 372, 515,
 516, 570, 590–595, 597–599,
 767, 768, 823, 843–848, 850–852
 \glossaryentrynumbers 11, 12, 187,
 218, 219, 233, 234, 238, 239,
 388, 389, 485, 504, 511, 513,
 514, 587, 638, 737, 756, 762,
 763, 765, 766, 840, 892, 991,
 1011, 1012, 1018, 1021, 1047,
 1147, 1167, 1168, 1174, 1176, 1177
 \glossaryheader 187,
 232, 241, 284, 288, 289, 291–
 296, 298, 299, 302–307, 309–
 312, 314, 316, 317, 321, 323,
 325, 328, 330–332, 334, 336,
 485, 510, 515, 587, 606, 608–
 613, 615–624, 626–628, 630–
 634, 736, 761, 767, 840, 859,
 861–867, 869–878, 880–888,
 991, 1017, 1022, 1147, 1173, 1178
 \glossarymark
 . 55, 55, 410, 660, 661, 916, 1071
 \glossaryname 19, 47, 49, 50,
 392, 407, 408, 642, 657, 658,
 896, 912–914, 1051, 1067–1069
 \glossarypostamble
 54, 187, 232, 410, 485,
 510, 587, 660, 736, 762, 840,
 916, 991, 1017, 1070, 1147, 1173
 \glossarypreamble
 54, 186, 232, 410, 484,
 509, 587, 660, 736, 761, 840,
 915, 991, 1017, 1070, 1146, 1172
 \glossarysection
 54, 186, 232, 410, 484,
 509, 587, 660, 736, 761, 840,
 916, 991, 1017, 1070, 1146, 1172
 \glossarystyle 246, 518, 770
 glossarysubentry (counter) 15
 \glossarysubentryfield
 243, 372, 516, 517,
 570, 590–599, 768, 823, 843–852
 \glossarytitle
 186, 218, 232, 236, 484, 504,
 509, 512, 587, 736, 755, 756,
 761, 764, 840, 991, 1011, 1012,
 1017, 1019, 1146, 1167, 1172, 1175
 \glossarytoctitle 11, 19, 20, 40,
 41, 47, 50, 54, 186, 218, 232,
 236, 237, 388, 392, 393, 404,
 405, 407, 408, 410, 484, 504,
 509, 512, 513, 587, 638, 642,
 643, 655, 657, 658, 660, 736,
 755, 761, 764, 765, 840, 892,
 896, 897, 910, 912, 914, 916,
 991, 1011, 1017, 1020, 1047,
 1051, 1052, 1065, 1067, 1069,
 1070, 1146, 1167, 1172, 1175, 1176
 \glossentry 104, 219, 234, 241, 284,
 288–291, 293, 294, 303, 304,
 306, 314, 316, 318, 321, 323,
 325, 329, 330, 332, 334, 440,
 504, 511, 516, 606, 608–611,
 615–617, 622–624, 626–628,
 630–633, 691, 756, 763, 768,
 859, 861–865, 869–871, 876–
 878, 880–885, 887, 947, 1012,
 1018, 1022, 1102, 1167, 1174, 1178
 \Glossentrydesc
 138, 242, 372, 516, 569,
 768, 822, 1023, 1123, 1178, 1179
 \glossentrydesc . . 138, 242, 242,
 285, 288–291, 293–295, 303,
 304, 306, 314, 316, 318, 321–
 323, 325, 329, 330, 332, 334,
 335, 372, 516, 569, 607–611,
 615–617, 622–624, 626–628,
 630–634, 767, 822, 860–865,
 869–871, 876–878, 880–882,
 884–887, 1023, 1123, 1178, 1179
 \Glossentryname 138,
242, 515, 767, 1023, 1123, 1178
 \glossentryname . . 138, 241, 241,
242, 243, 285, 287–291, 293,
 294, 303, 304, 306, 314, 316,
 318, 321, 323, 325, 329, 330,
 332, 334, 335, 371, 372, 515,

569, 606–611, 615–617, 622–
624, 626–628, 630–634, 767,
822, 860–865, 869–871, 876–
878, 880–882, 884, 885, 887,
1022, 1023, 1123, 1178, 1179
\glossentrynameother 138, 1123
\Glossentrysymbol
. 138, 242, 372, 516,
569, 768, 823, 1023, 1123, 1179
\glossentrysymbol 138, 242,
242, 285, 294, 295, 306, 318,
325, 329, 330, 332, 334, 335,
372, 516, 569, 606, 607, 611,
617, 624, 628, 630–634, 768,
823, 860, 865, 871, 878, 882,
884, 885, 887, 1023, 1123, 1179
\GLS 142, 464, 715, 969, 1125
\Gls 115, 141, 141,
250, 269, 445, 463, 520, 532,
696, 715, 772, 784, 953, 969,
1026, 1039, 1108, 1124, 1182, 1194
\gls 39, 114,
140, 141, 200, 240, 250, 269,
404, 445, 463, 494, 514, 520,
532, 654, 696, 715, 746, 766,
772, 784, 909, 953, 968, 969,
1001, 1022, 1026, 1039, 1064,
1108, 1124, 1157, 1177, 1182, 1194
\gls@delimN 248, 1025, 1181
\gls@accessibility
. 346, 347, 549, 802, 803
\gls@accsupp@engine 346, 346, 549, 802
\gls@Alphpage
. 207, 207, 211, 499, 500,
750–752, 1006, 1008, 1161–1163
\gls@alphpage
. 207, 207, 211, 499, 500,
750–752, 1006, 1008, 1161–1163
\gls@arabicpage
. 207, 207, 211, 499, 500,
750–752, 1006, 1007, 1161–1163
\gls@assign@desc 97, 97, 103, 435,
439, 686, 690, 942, 946, 1097, 1100
\gls@assign@desc@field 24
\gls@assign@descplural 533, 540,
542, 544, 578, 581, 582, 785,
792, 793, 795, 797, 832, 834, 835
\gls@assign@descplural@field . . 25
\gls@assign@field
. 86, 86, 91, 92, 97, 99,
101–103, 339, 428, 431, 435,
436, 438, 439, 601, 678, 682,
686–690, 854, 934, 938, 942–
946, 1089, 1092, 1093, 1097–1100
\gls@assign@firstpl 533–536, 538,
540, 542–544, 578–582, 784,
785, 787–793, 795, 797, 831–835
\gls@assign@name@field 25
\gls@assign@plural 533–536,
538, 540, 542–544, 578–582,
785, 787–793, 795, 797, 832–835
\gls@assign@symbolplural
. 533–536, 538,
542–544, 579, 580, 582, 785,
787–791, 795, 797, 832, 833, 835
\gls@assign@type@field 24
\gls@automake@makegloss
. 36, 37, 38, 197,
907, 908, 1000, 1062, 1063, 1155
\gls@automake@nr 36, 37,
38, 44, 197, 402, 493, 653, 744,
907, 908, 999, 1062, 1063, 1155
\gls@automake@option 36,
36, 37, 38, 907, 908, 1062, 1063
\gls@automake@val 36–
38, 402, 653, 907, 908, 1062, 1063
\gls@begindocdefs
. 88, 88, 429, 680, 936, 1090
\gls@checkisacronymlist
. 22, 126, 394,
454, 643, 705, 898, 958, 1052, 1113
\gls@checkseeallowed
. 81, 81, 198, 200, 425,
428, 493, 494, 675, 745, 746,
931, 1000, 1002, 1086, 1156, 1157
\gls@checkseeallowed@preambleonly
. 81, 425, 428, 675, 931, 1086
\gls@ccodepage 34, 66,
195, 196, 221, 401, 417, 491,
492, 505, 651, 667, 668, 743,
744, 757, 905, 923, 997, 998,
1013, 1060, 1078, 1153, 1154, 1168
\gls@debug@nr
. 7, 386, 404, 635, 654, 889, 1044
\gls@debug@val
. 7, 386, 404, 635, 654, 889, 1044
\gls@defdocnewglossaryentry . . .
. 86, 88, 112,
428, 429, 443, 679, 680, 694,
935, 936, 951, 1089, 1090, 1106

<code>\gls@defglossaryentry</code>	<code>\gls@inlinesubsep</code>
..... 86–89, 98, 98,	284,
428, 429, 435, 679, 680, 686,	285, 590, 606, 607, 843, 859, 860
935, 936, 942, 1089–1091, 1097	<code>\gls@islistofacronyms</code>
<code>\gls@delimN</code> 248, 248, 1025, 1181	21, 393, 394, 643, 897, 898, 1052
<code>\gls@delimR</code> 247, 248, 1025, 1181	<code>\gls@istfilebase</code>
<code>\gls@disablepagerefexpansion</code> 51, 51, 195, 196, 409,
206, 207, 211, 498–500, 750–	491, 492, 659, 743, 744, 914,
752, 1005, 1006, 1008, 1161, 1163	915, 997, 998, 1069, 1153, 1154
<code>\gls@do@addxdyattribute</code>	<code>\gls@kernelglossredefs@nr</code>
..... 60, 413, 663, 919, 1074 40, 909, 1064
<code>\gls@docdef@addlabel</code>	<code>\gls@kernelglossredefs@val</code>
. 87, 88, 679, 680, 935, 936, 1090 40, 909, 1064
<code>\gls@docdef@labellist</code>	<code>\gls@label</code> . 250, 520, 772, 1026, 1182
..... 87, 87, 679, 935, 1090	<code>\gls@level</code>
<code>\gls@docclearpage</code>	84,
56,	100, 101, 233, 427, 437, 510,
57, 411, 412, 661, 662, 917, 1072	511, 677, 688, 762, 933, 944,
<code>\gls@doentryfmt</code> 74, 422, 672, 928, 1083	1018, 1088, 1098, 1099, 1173, 1174
<code>\gls@dosubst</code>	<code>\gls@local@stepcounter</code>
... 130, 457, 707, 708, 961, 1116 107, 107, 948, 1103
<code>\gls@dotoctitle</code>	<code>\gls@measuring@reset</code>
218,	.. 107, 108, 948, 949, 1103, 1104
219, 236, 504, 512, 756, 764,	<code>\gls@mfirstuc@nr</code> . 41, 910, 911, 1065
1011, 1012, 1019, 1020, 1167, 1175	<code>\gls@mfirstuc@val</code> 41, 910, 1065
<code>\gls@end@sanitizesort</code>	<code>\gls@newacronym</code> 251, 251, 1027, 1182
25,	<code>\gls@noidx@nonumberlist</code>
26, 396, 645, 646, 900, 1054, 1055	.. 81, 203, 931, 1004, 1086, 1159
<code>\gls@endcheck</code>	<code>\gls@noidxglossary</code>
.... 85, 86, 428, 678, 934, 1089 200, 204, 494, 497,
<code>\gls@glossary</code> .. 39, 40, 205, 212–	746, 749, 1001, 1004, 1157, 1160
214, 404, 498, 501, 502, 654,	<code>\gls@nonumberlist@nr</code>
750, 753, 754, 909, 910, 1005, 81, 425, 675, 931, 1086
1008, 1009, 1064, 1160, 1164, 1165	<code>\gls@nonumberlist@val</code>
<code>\gls@glsdefs@addlabel</code> 81, 425, 675, 931, 1086
87, 89, 679, 680, 935, 936, 1090, 1091	<code>\gls@nosetquote</code>
<code>\gls@glsdefs@labellist</code>	98, 188,
..... 87, 87, 679, 935, 1090	190, 193, 435, 486, 487, 490,
<code>\gls@gobbleopt</code>	686, 737, 739, 742, 942, 992,
75, 76,	994, 996, 1097, 1148, 1149, 1152
422, 423, 672, 673, 928, 929, 1083	<code>\gls@number</code> . 211, 500, 752, 1007, 1163
<code>\gls@grplabel</code> ... 280, 605, 858, 1042	<code>\gls@numberedsection@nr</code>
<code>\gls@hypergroup prerun</code> 10, 237, 388, 512, 513, 638,
..... 283, 283, 605, 858, 1042	764, 892, 1020, 1046, 1175, 1176
<code>\gls@if@measuring</code>	<code>\gls@numberedsection@val</code>
... 106, 107, 108, 948, 949, 1103 10, 237, 388, 512,
<code>\gls@ifnotmeasuring</code> ... 107, 108,	638, 764, 892, 1020, 1046, 1175
109, 206, 441, 442, 692, 693,	<code>\gls@numberpage</code>
948–950, 1005, 1103–1105, 1161 207, 207, 211, 499, 500,
<code>\gls@inlinepostchild</code>	750–752, 1006, 1007, 1161–1163
284–	<code>\gls@org@glossaryentryfield</code> ...
286, 590, 606, 607, 843, 859, 860 219, 504, 756, 1012, 1167
<code>\gls@inlinesep</code>	
284,	
285, 590, 606, 607, 843, 859, 860	

<code>\gls@org@glossarysubentryfield</code>	<code>\gls@start@measuring</code>
. 219, 504, 756, 1012, 1167, 1168	107, 107, 108, 948, 949, 1103, 1104
<code>\gls@org@insert</code>	<code>\gls@stop@measuring</code>
537, 539, 541, 789, 791, 792, 794 108, 108, 949, 1103, 1104
<code>\gls@org@label</code>	<code>\gls@suffixF</code>
106–108, 948, 949, 1103	52, 52,
<code>\gls@org@link</code>	187, 189, 409, 485, 487, 587,
107, 108, 948, 949, 1103	589, 659, 737–739, 841, 842,
<code>\gls@org@refstepcounter</code>	915, 991–993, 1070, 1147, 1149
. . . 106, 107, 108, 948, 949, 1103	<code>\gls@suffixFF</code>
<code>\gls@org@target</code> 52, 53,
. 107, 108, 948, 949, 1103	187, 189, 409, 485, 487, 587–
<code>\gls@orgAlpha</code>	589, 659, 737, 739, 841, 842,
. 211, 500, 752, 1007, 1008, 1163	915, 991–993, 1070, 1147, 1149
<code>\gls@orgalph</code>	<code>\gls@text</code>
. 211, 500, 752, 1007, 1008, 1163	122, 452, 703, 957, 1111, 1112
<code>\gls@orgarabic</code>	<code>\gls@the</code>
. 211, 500, 752, 1007, 1163	. . . 211, 500, 752, 1007, 1163
<code>\gls@orgnumber</code>	<code>\gls@thissty</code>
. 211, 500, 752, 1007, 1163	. . 31, 400, 650, 904, 1059
<code>\gls@orgRoman</code>	<code>\gls@tmp</code>
. 211, 500, 752, 1007, 1008, 1163 197, 199, 204, 272,
<code>\gls@orgromannumeral</code>	497, 541, 749, 794, 999, 1001,
. 211, 500, 752, 1007, 1163	1004, 1039, 1155, 1156, 1160, 1195
<code>\gls@orgthe</code>	<code>\gls@tmplen</code>
. 211, 500, 752, 1007, 1163 137,
<code>\gls@original@glossary</code>	138, 333, 335, 462, 596, 632–
. . . 39,	634, 714, 849, 886, 887, 968, 1122
40, 403, 404, 653, 654, 909, 1064	<code>\gls@tr@set@acronym@toctitle</code>
<code>\gls@original@makeglossary</code> 20, 393, 642, 643, 897, 1051, 1052
. 39, 40,	<code>\gls@tr@set@main@toctitle</code>
403, 404, 653, 654, 909, 910, 1064 19, 392, 642, 896, 1051
<code>\gls@printglossary@begin</code>	<code>\gls@tr@set@numbers@toctitle</code>
. 218, 219, 1167, 1168 41, 405, 655, 910, 1065
<code>\gls@printglossary@end</code>	<code>\gls@tr@set@symbols@toctitle</code>
219, 220, 1168 40, 404, 655, 910, 1065
<code>\gls@protected@pagefmts</code>	<code>\gls@translate@nr</code>
130, 207, 30, 399, 649, 903, 1058
207, 208, 456, 499, 707, 750,	<code>\gls@translate@val</code>
751, 961, 1006, 1116, 1161, 1162 30, 399, 649, 903, 1058
<code>\gls@Romanpage</code>	<code>\gls@wrglossary</code>
. 207, 207, 211, 499, 500,	. . 206, 206, 498, 750, 1005, 1161
750–752, 1006, 1008, 1161–1163	<code>\gls@xdy@set@lang</code>
<code>\gls@romanpage</code> 33, 195, 196, 220, 905, 997,
. 207, 207, 211, 499, 500,	998, 1012, 1060, 1153, 1154, 1168
750–752, 1006, 1007, 1161–1163	<code>\gls@xdystring</code>
<code>\gls@save@numberlist</code> 130, 456, 457, 707, 708, 961, 1116
. . 11, 12, 216, 388, 389, 503,	<code>\gls@xindy@glsnumbersfalse</code>
638, 754, 892, 1010, 1047, 1166 34, 401, 651, 906, 1060
<code>\gls@seen@index@nr</code>	<code>\gls@xindy@glsnumberstrue</code>
. 9, 387, 637, 891, 1046 33, 401, 651, 905, 1060
<code>\gls@seen@index@val</code>	<code>\gls@xr@key</code>
. 9, 387, 637, 891, 1046 9, 81, 81, 387, 388, 425, 637,
<code>\gls@set@xr@key</code>	675, 891, 931, 1046, 1085, 1086
. 80, 80, 425, 675, 931, 1085	<code>\glsaccessdisplay</code>
 358, 557, 810
	<code>\glsaccessibility</code>
 346, 355, 549, 554, 802, 807

<code>\glsaccsupp</code>	354, 355, 355 , 554, 807, 808	<code>\glsbackslash</code> 44, 182 , 482, 734, 988, 1144
<code>\glsacronymtrue</code>	20, 393, 643, 897, 1052	<code>\glscapitalisewords</code> 175, 175 , 730, 984, 985, 1140
<code>\glsacrpluralsuffix</code>	<code>\glscapspace</code> 117, 118, 120,
..	48, 48 , 251, 261, 265–267,	121, 140–145, 161–169, 263,	264, 359–366, 376, 377, 446,
	407, 520, 526, 529–531, 533,	447, 449–451, 463–466, 472–	476, 528, 537, 541, 557, 558,
	657, 772, 778, 781–783, 786,	560–565, 573, 697, 698, 700–	702, 715–717, 723–728, 780,
	913, 1027, 1033, 1036–1038,	790, 794, 810, 811, 813–819,	826, 954–956, 968–971, 977–
	1067, 1182, 1188, 1189, 1192, 1193	982, 1035, 1109–1111, 1124–	1126, 1133–1137, 1190, 1191
<code>\glsacrshortcutsfalse</code>	<code>\glscategory</code> 354, 554, 807
.....	45, 406, 656, 911, 1066	<code>\glsclearpage</code> 56, 57 ,
<code>\glsacrshortcutstrue</code>	411, 412, 661, 662, 917, 1071, 1072	<code>\glsfclosebrace</code>
.....	45, 405, 656, 911, 1066	. 63, 182 , 187, 188,	416, 482, 485, 587 , 588, 666,
<code>\glsacspace</code>	734, 737, 840, 841, 921, 922,	988, 991, 992, 1076, 1144, 1147
	259, 260 , 262, 524, 525, 527,	<code>\glscompositor</code> 52, 52 , 62,
	776–778, 1031–1033, 1187, 1189	189, 409, 415, 487, 587, 589,	659, 665, 738, 840, 842, 915,
<code>\glsadd</code> 39, 181,	921, 993, 1069, 1070, 1076, 1149	<code>\glscounter</code>
	181 , 182, 404, 481, 482, 654,	<code>\glscounter</code> 22, 22 ,
	733, 909, 988, 1064, 1143, 1144	46, 58, 76, 102, 128, 394, 406,	412, 423, 438, 455, 584, 644,
<code>\glsaddall</code>	.. 181 , 482, 733, 988, 1144	656, 662, 673, 689, 706, 837,	898, 912, 918, 929, 945, 959,
<code>\glsaddallunused</code>	1053, 1066, 1073, 1083, 1100, 1114	<code>\glscurrententrylabel</code>
.....	182 , 482, 733, 988, 1144	<code>\glscurrententrylabel</code>
<code>\glsaddkey</code>	... 91 , 431, 682, 938, 1093	138, 216, 219, 503, 504, 755,	756, 1010, 1012, 1123, 1166–1168
<code>\GlsAddLetterGroup</code>	<code>\glscurrentfieldvalue</code>	... 72, 73,
.....	66 , 418, 668, 924, 1078	73 , 421, 422, 671, 672, 927, 1082	<code>\glscustomtext</code>
<code>\glsaddprotectedpagefmt</code>	117, 119, 121,	140–145, 161–169, 263, 264,
..	207 , 208, 499, 751, 1006, 1162	359, 361, 363–366, 376, 377,	446, 449–451, 463–466, 472–
<code>\GlsAddSortRule</code>	476, 528, 533, 534, 537, 539,	541, 557, 560, 562–565, 573,
.....	64 , 416, 417, 667, 922, 1077	697, 699–702, 715–717, 723–	728, 780, 786, 789–792, 794,
<code>\glsaddstoragekey</code>	810, 813, 815, 817–819, 826,	827, 954–956, 968–971, 977–
.....	90 , 431, 681, 937, 1092	982, 1034, 1035, 1109–1111,	1124–1126, 1133–1137, 1190, 1191
<code>\GlsAddXdyAlphabet</code>		
.....	61 , 414, 664, 920, 1075		
<code>\GlsAddXdyAttribute</code> 59,		
	59 , 60, 61, 413, 414, 584, 663,		
	664, 837, 838, 918–920, 1073–1075		
<code>\GlsAddXdyCounters</code>		
.....	58 , 59, 60, 77, 412, 413,		
	423, 584, 662–664, 673, 837,		
	918, 919, 929, 1073, 1074, 1084		
<code>\GlsAddXdyLocation</code>	. 63 , 415, 416,		
	584, 666, 838, 921, 922, 1076, 1077		
<code>\GlsAddXdyStyle</code>	64 , 417, 667, 923, 1077		
<code>\glsautomakefalse</code> 37,		
	197, 402, 403, 493, 653, 745,		
	908, 999, 1000, 1062, 1063, 1155		
<code>\glsautomaketrue</code>		
	37, 38, 402, 653, 908, 1062, 1063		
<code>\glsautoprefix</code>		
..	10 , 11, 237, 388, 513, 638,		
	765, 892, 1020, 1046, 1047, 1176		

<code>\GlsDeclareNoHyperList</code>	555–557, 582, 583, 601, 633,
. 22 , 23 , 394 , 644 , 898 , 1053	652, 669–671, 676, 680, 684–
<code>\glsdefaultshortaccess</code>	686, 690, 693–695, 705, 728,
. 382 , 382 , 578–581 , 831–834	732, 733, 746–748, 751, 752,
<code>\glsdefaulttype</code> 19 , 19 , 20 , 33 ,	756, 758, 759, 762–764, 766,
54 , 65 , 67 , 74 , 75 , 99 , 117 , 204 ,	767, 799–802, 808–810, 835,
217 , 218 , 392 , 410 , 417 , 418 ,	836, 854, 855, 886, 907, 924–
422 , 436 , 446 , 452 , 497 , 503 ,	927, 931, 932, 936, 940–942,
504 , 642 , 660 , 667 , 668 , 672 ,	947, 950–952, 959, 982, 983,
687 , 697 , 703 , 749 , 755 , 896 ,	987, 988, 1002, 1003, 1007,
905 , 915 , 923 , 924 , 928 , 943 ,	1008, 1012, 1014, 1015, 1018,
954 , 1004 , 1011 , 1051 , 1060 ,	1019, 1021, 1022, 1040, 1041,
1070 , 1078 , 1079 , 1082 , 1083 ,	1062, 1079–1082, 1086, 1091,
1098 , 1108 , 1109 , 1160 , 1166 , 1167	1094–1097, 1101, 1105–1107,
<code>\glsdefmain</code> 19 , 20 , 77 , 392 ,	1114, 1137, 1138, 1142, 1143,
423 , 642 , 673 , 896 , 929 , 1051 , 1084	1157–1159, 1163, 1167–1170,
<code>\glsdefs@newdocentry</code>	1173–1175, 1177, 1178, 1195–1197
. 88 , 89 , 680 , 936 , 1091	<code>\glsdisablehyper</code>
<code>\GLSdesc</code> 152 , 468 , 720 , 973 , 1129 139 , 463 , 714 , 968 , 1124
<code>\Glsdesc</code> 151 , 151 , 468 , 720 , 973 , 1129	<code>\Glsdisp</code> 146 , 146 , 971 , 1126
<code>\glsdesc</code> 151 , 151 , 468 , 720 , 973 , 1129	<code>\glsdisp</code> 145 , 146 , 466 , 717 , 971 , 1126
<code>\GLSdescplural</code> 153 , 469 , 720 , 974 , 1129	<code>\glsdisplay</code> 122 , 446 , 452 , 453 , 697 , 703
<code>\Glsdescplural</code>	<code>\glsdisplayfirst</code>
. 152 , 152 , 468 , 720 , 974 , 1129 122 , 446 , 452 , 697 , 703
<code>\glsdescplural</code>	<code>\glsdisplaynumberlist</code>
152 , 152 , 468 , 720 , 973 , 974 , 1129 179 , 202 , 480 , 495 ,
<code>\glsdescriptionaccessdisplay</code> . .	732 , 747 , 987 , 1003 , 1142 , 1158
. 356 , 368 ,	<code>\glsdohyperlink</code> 139 , 139 , 140 ,
372 , 555 , 558–560 , 567 , 569 ,	462 , 463 , 714 , 715 , 968 , 1123 , 1124
570 , 809 , 812 , 813 , 820 , 822 , 823	<code>\glsdohyperlinkhook</code> 139 , 139 , 1123
<code>\glsdescriptionpluralaccessdisplay</code>	<code>\glsdohypertarget</code> 137 , 139 , 140 ,
. 357 , 368 ,	462 , 463 , 714 , 715 , 968 , 1122 , 1124
555 , 557 , 558 , 567 , 809–811 , 820	<code>\glsdohypertargethook</code>
<code>\glsdescwidth</code> 291 , 291 , 292 , 293 , 138 , 138 , 1122 , 1123
296 , 297 , 299 , 300 , 302 , 302 ,	<code>\glsdoifexists</code> 69 , 70–72 , 94–97 ,
303–307 , 314 , 314 , 315–317 ,	108 , 109 , 140–146 , 161–169 ,
319–321 , 321 , 322–327 , 609–	175 , 179 , 181 , 201 , 202 , 341–
612 , 614–618 , 622–629 , 863 ,	345 , 364–366 , 371 , 372 , 419–
864 , 866–872 , 875–877 , 879–883	421 , 433–435 , 441 , 442 , 463–
<code>\glsdetoklabel</code> 35 , 68 , 68 , 69–73 ,	466 , 472–476 , 478 , 480 , 481 ,
82 , 89 , 94–98 , 104 , 109 , 110 ,	495 , 496 , 564 , 565 , 569 , 602–
112–114 , 127 , 169–171 , 179–	604 , 669–671 , 684–686 , 692 ,
181 , 200–202 , 209 , 212 , 219 ,	693 , 715–718 , 723–727 , 730 ,
227–229 , 233 , 235 , 239 , 240 ,	732 , 733 , 747 , 748 , 817 , 818 ,
242 , 274–278 , 333 , 339–341 ,	822 , 823 , 855–858 , 924–927 ,
355–358 , 383 , 384 , 418–421 ,	940–942 , 949 , 950 , 968–971 ,
426 , 429 , 433–435 , 440 , 442–	977–982 , 985–988 , 1002 , 1003 ,
444 , 454 , 476 , 477 , 480 , 481 ,	1079–1081 , 1094–1096 , 1104 ,
494–496 , 500 , 504 , 506 , 507 ,	1105 , 1124–1127 , 1133–1137 ,
510–512 , 514 , 515 , 546–549 ,	1140 , 1142 , 1143 , 1158 , 1159

`\glsoifexistsordo` . 69, 125, 170, 419, 454, 476, 669, 704, 728, 925, 958, 982, 1080, 1113, 1138
`\glsoifexistsorwarn` ... 69, 235, 241, 242, 419, 512, 515, 516, 669, 764, 767, 768, 925, 1019, 1022, 1023, 1079, 1175, 1178, 1179
`\glsoifnoexists` 69, 86, 97, 419, 428, 435, 669, 678, 686, 925, 934, 942, 1079, 1089, 1097
`\glsdonohyperlink` 127, 139, 139, 455, 462, 463, 706, 714, 959, 968, 1114, 1123, 1124
`\glsdoparenifnotempty` 272, 541, 793, 1039, 1195
`\glsdosanitizesort` 16, 390, 391, 640, 894, 1049
`\glsdshowtarget` 8, 137, 139, 891, 968, 1045, 1122, 1123
`\glsenableentrycount` .. 111, 443, 444, 694, 695, 951, 952, 1106, 1107
`\glsenablehyper` 140, 463, 714, 968, 1124
`\glscapwrcontent` 127, 128, 959, 1114
`\glsendryaccess` 351, 356, 553, 555, 806, 808
`\glsendrycounter` . 128, 245, 248, 455, 517, 519, 706, 769, 771, 959, 1024, 1026, 1114, 1180, 1181
`\glsendrycounterfalse` 14, 390, 640, 894, 1048
`\glsendrycounterlabel` 240, 240, 514, 515, 766, 767, 1022, 1177, 1178
`\GlsEntryCounterLabelPrefix` ... 237, 239, 240, 513, 514, 765, 766, 1020–1022, 1176, 1177
`\glsendrycountertrue` 14, 390, 640, 894, 1048
`\glsendrycurrcount` 112–114, 443–445, 694–696, 951–953, 1106, 1107
`\Glsentrydesc` . 152, 172, 173, 242, 368, 372, 468, 477, 516, 567, 569, 720, 729, 768, 820, 822, 973, 983, 1023, 1129, 1139, 1179
`\glsendrydesc` 138, 151, 152, 172, 242, 368, 372, 447–449, 468, 477, 516, 558–560, 567, 569, 698–700, 720, 729, 767, 812, 813, 820, 822, 973, 983, 1023, 1123, 1129, 1139, 1178
`\glsendrydescaccess` 352, 357, 553, 555, 806, 809
`\Glsentrydescplural` 153, 173, 173, 368, 469, 478, 567, 720, 729, 820, 974, 984, 1129, 1139
`\glsendrydescplural` 152, 153, 173, 368, 446, 447, 468, 469, 477, 557, 558, 567, 697, 698, 720, 729, 810, 811, 820, 974, 984, 1129, 1139
`\glsendrydescpluralaccess` 352, 357, 553, 556, 806, 809
`\Glsentryfirst` 115, 119, 148, 174, 174, 360, 367, 445, 448, 450, 467, 478, 559, 561, 566, 696, 699, 701, 718, 730, 812, 815, 819, 953, 955, 972, 984, 1108, 1110, 1127, 1140
`\glsendryfirst` 114, 119, 147, 148, 174, 360, 361, 367, 445, 447, 448, 450, 467, 478, 559, 561, 562, 566, 696, 698, 699, 701, 718, 730, 812–815, 819, 953, 955, 972, 984, 1108–1110, 1127, 1140
`\glsendryfirstaccess` 352, 356, 553, 555, 806, 808
`\Glsentryfirstplural` 116, 118, 150, 175, 175, 359, 367, 446, 447, 449, 467, 478, 558, 561, 566, 697, 698, 700, 719, 730, 811, 814, 819, 954, 973, 984, 1108, 1109, 1128, 1140
`\glsendryfirstplural` 116, 118, 149, 150, 174, 359, 360, 367, 445–447, 449, 467, 468, 478, 557, 558, 560, 561, 566, 696–698, 700, 719, 730, 810, 811, 813, 814, 819, 820, 953, 954, 972, 973, 984, 1108, 1109, 1128, 1140
`\glsendryfirstpluralaccess` 352, 356, 553, 555, 806, 808
`\glsendryfmt` 76, 78, 117, 423, 446, 673, 674, 697, 929, 954, 1083, 1084, 1109
`\Glsentryfull` 178, 178, 257, 265, 267, 378, 380, 480, 523, 529, 530, 574, 576, 732, 775,

781, 782, 827, 829, 986, 1029, 1036, 1037, 1142, 1185, 1191, 1193	265, 364, 373, 376, 378, 446, 476, 480, 524, 525, 528, 529, 564, 570, 573, 574, 697, 727, 731, 732, 776, 777, 780, 781, 817, 823, 826–828, 954, 981, 986, 1031, 1035, 1036, 1108, 1137, 1142, 1186, 1187, 1190–1192
<code>\glsentryfull</code> 178, 257, 265, 267, 378, 380, 480, 523, 529, 530, 574, 576, 731, 775, 781, 782, 827, 829, 986, 1029, 1036, 1037, 1142, 1185, 1191, 1193	<code>\glsentrylongpl</code> 116, 122, 167, 169, 178, 178, 259, 260, 264–267, 364, 373, 374, 376, 378–380, 445, 452, 475, 476, 480, 524, 525, 528–531, 539, 545, 564, 570, 571, 573–576, 696, 703, 727, 728, 731, 732, 776, 777, 780–782, 792, 798, 817, 823, 824, 826–829, 953, 957, 981, 982, 986, 1031, 1032, 1035–1037, 1108, 1112, 1136, 1137, 1142, 1186–1188, 1190–1193
<code>\Glsentryfullpl</code> . . . 179, 179, 257, 265, 267, 378, 380, 480, 523, 529, 531, 574, 576, 732, 775, 781, 782, 828, 829, 986, 1030, 1036, 1037, 1142, 1185, 1192, 1193	<code>\Glsentryname</code> 151, 171, 171, 242, 368, 372, 468, 477, 515, 567, 569, 719, 728, 767, 820, 822, 973, 982, 983, 1023, 1129, 1138, 1178
<code>\glsentryfullpl</code> 178, 257, 265, 267, 378, 380, 480, 523, 529, 531, 574, 576, 732, 775, 781, 782, 827, 829, 986, 1030, 1036, 1037, 1142, 1185, 1191, 1193	<code>\glsentryname</code> 138, 150, 151, 170, 215, 333, 368, 371, 468, 477, 566, 567, 569, 633, 719, 720, 728, 820, 822, 886, 973, 982, 983, 1010, 1123, 1128, 1129, 1138, 1166
<code>\glsentryitem</code> . 240, 285, 287–291, 293, 294, 303, 304, 306, 314, 316, 318, 321, 323, 325, 329, 330, 332, 334, 515, 590–595, 597–599, 606, 608–611, 615– 617, 622–624, 626–628, 630– 633, 767, 843–852, 859, 861– 865, 869–871, 876–878, 880– 882, 884, 885, 887, 1022, 1178	<code>\glsentrynumberlist</code> 179, 179, 201, 480, 495, 732, 747, 986, 987, 1002, 1142, 1158
<code>\Glsentrylong</code> 115, 166, 172, 178, 178, 259, 264, 265, 364, 366, 373, 377, 378, 445, 474, 477, 480, 524, 528, 529, 564, 565, 570, 573, 574, 696, 726, 728, 731, 732, 776, 780, 781, 817, 818, 823, 826, 827, 953, 980, 983, 986, 1031, 1035, 1036, 1108, 1136, 1139, 1142, 1186, 1187, 1191	<code>\glsentryparent</code> 175, 479, 730, 985, 1140
<code>\glsentrylong</code> 114, 121, 165, 167, 172, 178, 178, 258– 268, 363, 365, 366, 373–382, 445, 452, 474, 475, 477, 480, 524–532, 539, 563, 565, 570– 577, 696, 702, 726, 728, 731, 732, 776–784, 792, 817–819, 823–831, 953, 957, 980, 983, 986, 1031–1038, 1108, 1111, 1135, 1136, 1138, 1142, 1186–1194	<code>\Glsentryplural</code> 118, 149, 173, 173, 359, 367, 447, 449, 467, 478, 557, 560, 566, 697, 700, 719, 729, 811, 814, 819, 954, 972, 984, 1109, 1128, 1139
<code>\glsentrylongaccess</code> 353, 357, 553, 556, 806, 809	<code>\glsentryplural</code> . . . 118, 148, 149, 173, 359, 360, 367, 446, 447, 449, 467, 478, 557, 558, 560, 561, 566, 697, 698, 700, 719, 729, 810, 811, 813, 814, 819, 954, 972, 984, 1109, 1128, 1139
<code>\Glsentrylongpl</code> 116, 168, 178, 178, 179, 259, 264,	<code>\glsentrypluralaccess</code> 352, 356, 553, 555, 806, 808

<code>\Glsentryprefix</code>	<code>\glsentryshortpl</code> 120, 122, 163, 165,
340, <u>340</u> , 343, 601, 603, 854, 856	<u>177</u> , 178, 179, 259, 260, 265–
<code>\glsentryprefix</code> .. <u>339</u> , 340, 341,	267, 361, 362, 364, 373, 374,
344, 601, 602, 604, 854, 855, 857	378–380, 450, 452, 473, 474,
<code>\Glsentryprefixfirst</code>	480, 524, 525, 529–531, 545,
<u>339</u> , 340, 343, 601, 603, 854, 856	562, 564, 570, 571, 574–576,
<code>\glsentryprefixfirst</code> <u>339</u> , 340, 341,	701, 703, 725, 731, 732, 776,
344, 601, 602, 604, 854, 855, 857	777, 781, 782, 798, 815, 817,
<code>\Glsentryprefixfirstplural</code>	823, 824, 827–829, 955, 957,
340, <u>340</u> , 344, 601, 604, 854, 857	979, 986, 1031, 1032, 1036,
<code>\glsentryprefixfirstplural</code>	1037, 1110, 1112, 1134, 1135,
. <u>339</u> , 340, 342,	1141, 1142, 1186–1188, 1191–1193
345, 601, 602, 605, 854, 856, 858	<code>\glsentryshortpluralaccess</code>
<code>\Glsentryprefixplural</code> <u>352</u> , 357, 553, 556, 806, 809
340, <u>340</u> , 343, 601, 603, 854, 857	<code>\glsentrysort</code> <u>175</u> , 479, 730, 985, 1140
<code>\glsentryprefixplural</code>	<code>\Glsentrysymbol</code> 154, 174, <u>174</u> , 242,
. <u>339</u> , 340, 342,	369, 372, 469, 478, 516, 567,
345, 601, 602, 604, 854, 856, 858	570, 721, 729, 768, 820, 823,
<code>\glsentryprevcount</code> 112, 113, 443,	974, 984, 1023, 1130, 1140, 1179
444, 694, 695, 951, 952, 1106, 1107	<code>\glsentrysymbol</code> .. 138, 153, 154,
<code>\Glsentryshort</code> 120, 162,	<u>174</u> , 242, 369, 372, 447–449,
172, 177, <u>177</u> , 260, 266, 267,	469, 478, 516, 534, 537, 541,
362, 364, 374, 379, 380, 451,	558–560, 567, 569, 698–700,
472, 477, 479, 525, 530, 563,	720, 721, 729, 768, 786, 790,
564, 571, 575, 576, 702, 724,	794, 812, 813, 820, 823, 974,
729, 731, 777, 782, 816, 817,	984, 1023, 1123, 1130, 1139, 1179
824, 828, 829, 956, 978, 983,	<code>\glsentrysymbolaccess</code>
986, 1032, 1036, 1037, 1111, <u>352</u> , 356, 553, 555, 806, 808
1133, 1139, 1141, 1188, 1192, 1193	<code>\Glsentrysymbolplural</code> 155,
<code>\glsentryshort</code> 120, 122,	174, <u>174</u> , 369, 469, 478, 567,
161, 163, 172, <u>177</u> , 178, 257–	721, 730, 821, 975, 984, 1130, 1140
268, 362–365, 373–382, 451,	<code>\glsentrysymbolplural</code>
452, 472, 473, 477, 479, 480, 154, 155, <u>174</u> , 369,
523–532, 562–565, 570–577,	446, 447, 469, 478, 534, 537,
702, 723, 724, 729, 731, 732,	541, 557, 558, 567, 697, 698,
775–784, 816–818, 823–831,	721, 730, 786, 790, 794, 810,
956, 957, 977, 978, 983, 986,	811, 821, 974, 975, 984, 1130, 1140
1030–1038, 1110, 1111, 1133,	<code>\glsentrysymbolpluralaccess</code>
1134, 1139, 1141, 1142, 1185–1194 <u>352</u> , 356, 553, 555, 806, 809
<code>\glsentryshortaccess</code>	<code>\Glsentrytext</code> 119, 147,
. . . . <u>352</u> , 357, 553, 556, 806, 809	173, <u>173</u> , 360, 366, 448, 450,
<code>\Glsentryshortpl</code> 120,	466, 478, 559, 561, 566, 699,
164, 178, <u>178</u> , 260, 266, 267,	701, 718, 729, 812, 814, 819,
362, 374, 379, 380, 450, 473,	955, 971, 984, 1110, 1127, 1139
480, 525, 530, 531, 562, 571,	<code>\glsentrytext</code> 118,
575, 576, 701, 725, 731, 777,	119, 146, 147, <u>173</u> , 180, 215,
782, 815, 824, 828, 829, 955,	360, 361, 366, 447, 448, 450,
979, 986, 1032, 1037, 1110,	466, 478, 481, 503, 558, 559,
1135, 1141, 1142, 1188, 1192, 1193	561, 562, 566, 698–701, 718,
	729, 733, 754, 812, 814, 815,

819, 955, 971, 984, 987, 1010, 1109, 1110, 1127, 1139, 1143, 1166	723, 731, 822, 977, 986, 1133, 1141
<code>\glsentrytextaccess</code>	<code>\glsentryuservi</code> 160, 161, <u>177</u> , 371, 471, 472, 479, 569, 723, 731, 822, 977, 986, 1132, 1133, 1141
. <u>351</u> , 356, 553, 555, 806, 808	<code>\glsentryuserviaccess</code>
<code>\glsentrytitlecase</code> <u>353</u> , 358, 554, 557, 807, 810
. <u>175</u> , 478, 479, 730, 985, 1140	<code>\glsesclocationsfalse</code> . . 12, 200, 494, 746, 893, 1001, 1047, 1157
<code>\glsentrytype</code> 99, <u>175</u> , 436, 479, 687, 730, 943, 985, 1098, 1140	<code>\glsesclocationstrue</code> 389, 638
<code>\Glsentryuseri</code> 156, 176, <u>176</u> , 369, 470, 479, 568, 721, 731, 821, 975, 985, 1131, 1141	<code>\glsexpandfields</code>
<code>\glsentryuseri</code> 155, 156, <u>176</u> , 369, 469, 470, 479, 568, 721, 730, 821, 975, 985, 1130, 1131, 1140 <u>86</u> , 428, 678, 934, 1089
<code>\glsentryuseriaccess</code>	<code>\glsfieldaccsupp</code>
. <u>353</u> , 357, 553, 556, 807, 809	354, <u>354</u> , 355, 554, 555, 807, 808
<code>\Glsentryuserii</code> 157, 176, <u>176</u> , 370, 470, 479, 568, 722, 731, 821, 975, 985, 1131, 1141	<code>\glsfielddef</code> . . <u>95</u> , 434, 684, 940, 1095
<code>\glsentryuserii</code> 156, 157, <u>176</u> , 370, 470, 479, 568, 721, 722, 731, 821, 975, 985, 1131, 1141	<code>\glsfielddedef</code> . <u>94</u> , 433, 684, 940, 1095
<code>\glsentryuseriiaccess</code>	<code>\glsfieldfetch</code> <u>95</u> , 175, 434, 478, 685, 730, 941, 985, 1095, 1140
. <u>353</u> , 358, 553, 556, 807, 809	<code>\glsfieldgdef</code> . . <u>94</u> , 433, 684, 940, 1095
<code>\Glsentryuseriii</code> 158, 176, <u>176</u> , 370, 470, 479, 568, 722, 731, 821, 976, 985, 1131, 1141	<code>\glsfieldxdef</code> . . <u>94</u> , 433, 684, 940, 1094
<code>\glsentryuseriii</code> 157, 158, <u>176</u> , 370, 470, 479, 568, 722, 731, 821, 976, 985, 1131, 1141	<code>\glsfindwidesttoplevelname</code>
<code>\glsentryuseriiiaccess</code> <u>333</u> , 632, 886
. <u>353</u> , 358, 554, 556, 807, 809	<code>\GLSfirst</code> . . . <u>148</u> , 467, 718, 972, 1127
<code>\Glsentryuseriv</code> 159, 177, <u>177</u> , 370, 471, 479, 568, 722, 731, 821, 976, 985, 1132, 1141	<code>\Glsfirst</code> <u>147</u> , <u>147</u> , 467, 718, 972, 1127
<code>\glsentryuseriv</code> 158, 159, <u>176</u> , 370, 471, 479, 568, 722, 731, 821, 822, 976, 985, 1132, 1141	<code>\glsfirst</code>
<code>\glsentryuserivaccess</code>	<u>147</u> , <u>147</u> , 467, 718, 971, 972, 1127
. <u>353</u> , 358, 554, 556, 807, 810	<code>\glsfirstaccessdisplay</code>
<code>\Glsentryuserv</code> 160, 177, <u>177</u> , 371, 471, 479, 568, 723, 731, 822, 977, 986, 1132, 1141	<u>356</u> , 360, 361, 367, 555, 559, 561, 562, 566, 808, 812–815, 819
<code>\glsentryuserv</code> 159, 160, <u>177</u> , 371, 471, 479, 568, 569, 723, 731, 822, 976, 977, 985, 986, 1132, 1141	<code>\GLSfirstplural</code>
<code>\glsentryuservaccess</code> <u>150</u> , 467, 719, 973, 1128
. <u>353</u> , 358, 554, 556, 807, 810	<code>\Glsfirstplural</code>
<code>\Glsentryuservi</code> 161, 177, <u>177</u> , 371, 471, 479, 569,	. . . 149, <u>149</u> , 467, 719, 972, 1128
	<code>\glsfirstplural</code>
	. . . 149, <u>149</u> , 467, 719, 972, 1128
	<code>\glsfirstpluralaccessdisplay</code> . .
 <u>356</u> , 359, 367, 555, 557, 558, 560, 561, 566, 808, 810, 811, 813, 814, 819, 820
	<code>\glsfirstpluralacessdisplay</code> . .
 360, 561, 814
	<code>\glsgenacfmt</code> <u>119</u> , 258–260, 266, <u>361</u> , 373, 379, 450, 524, 525, 530, 562, 570, 571, 575, 701, 776, 777, 781, 815, 823, 824, 828, 955, 1030– 1032, 1036, 1110, 1186, 1187, 1192
	<code>\glsgenentryfmt</code>
	. 117, <u>117</u> , 258–260, 264, 266, 270, <u>359</u> , 373, 377, 379, 449, 524, 525, 528, 530, 532, 534,

535, 537, 539, 541, 543, 545, 560, 570, 571, 573, 575, 700, 776, 777, 780, 781, 784, 786, 788, 789, 792, 794, 796, 798, 813, 823, 824, 827, 828, 954, 1030–1032, 1035, 1036, 1039, 1109, 1186, 1187, 1191, 1192, 1195	<code>\glsifhyper</code> 124 , 453, 704
<code>\glsgetgrouplabel</code> 245 , 517, 769, 1024, 1179	<code>\glsifhyperon</code> 126 , 453, 454, 704, 705, 959, 1113
<code>\glsgetgrouptitle</code> 244 , 284, 287–289, 308–313, 329–333, 336, 517, 606, 608, 609, 619–622, 630–632, 634, 769, 859, 861, 862, 872–875, 884–886, 888, 1023, 1043, 1179	<code>\glsIfListOfAcronyms</code> 21, 21 , 22, 393, 394, 643, 897, 898, 1052
<code>\gls glossarymark</code> 54 , 54 , 410, 660, 916, 1070, 1071	<code>\glsifmeasuring</code> 107, 107 , 108, 948, 949, 1103
<code>\gls groupheading</code> 188, 234, 244 , 284, 288, 289, 291, 293, 294, 302, 304, 306, 308–314, 316, 318, 321, 323, 325, 328–334, 336, 485, 511, 517, 588, 606, 608–611, 615– 617, 619–624, 626–628, 630– 634, 737, 763, 769, 841, 859, 861–865, 869–878, 880–888, 992, 1018, 1023, 1147, 1174, 1179	<code>\glsifplural</code> 117, 120, 140–145, 161–169, 263, 359, 361, 362, 364–366, 376, 446, 449–451, 463–466, 472– 476, 528, 534, 537, 539, 541, 557, 560, 562–565, 573, 697, 700–702, 715–717, 723–728, 780, 786, 789, 792, 794, 810, 813, 815–819, 826, 954–956, 968–971, 977–982, 1035, 1109– 1111, 1124–1126, 1133–1137, 1190
<code>\gls groupskip</code> 187, 188, 234, 243 , 285, 288, 292, 293, 295, 298, 299, 303–306, 315, 316, 318, 322, 324, 326, 329, 331, 332, 335, 485, 511, 517, 587, 588, 607, 608, 610, 611, 613, 616, 617, 622–624, 626– 628, 630–632, 634, 737, 763, 769, 840, 841, 860, 861, 863– 867, 869–871, 876–878, 880– 882, 884–886, 888, 991, 992, 1018, 1023, 1147, 1174, 1179	<code>\glsifusedtranslatordict</code> 29 , 399, 649, 903, 1058
<code>\glshyperfirstfalse</code> 266, 379, 530, 575, 782, 828, 1036, 1192	<code>\glsifusetranslator</code> 29–31, 49, 50, 337, 399, 400, 407, 408, 583, 600, 648–650, 657, 658, 836, 853, 903, 904, 913, 914, 1057–1059, 1068, 1069
<code>\glshyperfirsttrue</code> 31, 400, 650, 904, 1059	<code>\glsignore</code> . . . 182 , 482, 733, 988, 1144
<code>\glshyperlink</code> . 180 , 215, 481, 503, 733, 754, 987, 1010, 1143, 1166	<code>\glsindexingsetting</code> 15 , 18, 197, 200, 894, 896, 1000, 1001, 1049–1051, 1155, 1157
<code>\glshypernavsep</code> 283 , 283 , 284, 606, 859, 1043	<code>\glsindexonlyfirstfalse</code> 31, 400, 650, 904, 1059
<code>\glshypernumber</code> . . . 53, 247 , 249, 409, 518–520, 660, 770–772, 915, 1025, 1026, 1070, 1180, 1182	<code>\glsinlinedescformat</code> 285, 286 , 590, 607, 843, 860
	<code>\glsinlinedopostchild</code> 284, 285 , 590, 606, 607, 843, 859, 860
	<code>\glsinlineemptydescformat</code> 285, 286 , 590, 606, 607, 843, 860
	<code>\glsinlineifhaschildren</code> . . . 285, 285
	<code>\glsinlinenameformat</code> 285, 286 , 590, 606, 607, 843, 860
	<code>\glsinlineparentchildseparator</code> 285, 286 , 590, 607, 843, 860
	<code>\glsinlinepostchild</code> 285, 286 , 590, 607, 843, 860
	<code>\glsinlineseparator</code> 285, 286 , 590, 607, 843, 860
	<code>\glsinlinesubdescformat</code> 285, 286 , 590, 607, 843, 860, 861
	<code>\glsinlinesubnameformat</code> 285, 286 , 590, 607, 843, 860, 861

<code>\glsinlinesubseparator</code>	829, 958, 1029, 1035–1037, 1039, 1113, 1185, 1191–1193, 1195
<code>\glsinsert</code>	118–121, 140–145, 161–169, 264, 359–366, 376, 377, 446–451, 463–466, 472–476, 528, 534, 537, 539, 541, 557–565, 573, 697–702, 715–717, 723–728, 780, 786, 789–792, 794, 810–819, 826, 827, 954–956, 969–971, 977–982, 1035, 1109–1111, 1124–1126, 1133–1137, 1190, 1191
<code>\glskeylisttok</code>	255, 256, 382, 522, 533–536, 538, 540, 542–546, 578–582, 774, 784, 785, 787–793, 795–797, 799, 831–835, 1028, 1029, 1184, 1185
<code>\glslabel</code>	98, 117–121, 126, 127, 162–169, 258–260, 263, 264, 266, 359–363, 373, 376, 377, 379, 436, 446–451, 454, 455, 472–476, 524, 525, 528, 530, 534, 537, 539, 541, 557–563, 570, 571, 573, 575, 686, 697–702, 705, 706, 724–728, 776, 777, 780, 781, 786, 789, 790, 792, 794, 810–816, 823, 824, 826, 828, 942, 954–956, 958, 959, 978–982, 1030–1032, 1035, 1036, 1097, 1109–1111, 1113, 1114, 1133–1137, 1186, 1187, 1190–1192
<code>\glslabelhypertarget</code>	138, 1123
<code>\glslabelhypertargetdefs</code>	138, 138, 1123
<code>\glslabelhypertargetprefix</code>	138, 138, 1123
<code>\glslabelhypertargetvalue</code>	138, 138, 1123
<code>\glslabeltok</code>	255, 256, 522, 532–546, 578–581, 774, 784–794, 796–799, 831–834, 1028, 1029, 1184, 1185
<code>\glsletentryfield</code>	169, 476, 728, 982, 1137
<code>\Glslink</code>	125, 126, 958, 1113
<code>\glslink</code> 125, 126, 256, 257, 264–266, 270, 377–380, 453, 522, 523, 529, 530, 533, 574–576, 704, 774, 775, 780–782, 786, 827,	829, 958, 1029, 1035–1037, 1039, 1113, 1185, 1191–1193, 1195
<code>\glslinkcheckfirsthyperhook</code> ...	126, 126, 454, 705, 958, 959, 1113
<code>\glslinkpostsetkeys</code>	126, 127, 454, 455, 705, 959, 1113, 1114
<code>\glslinkvar</code>	124, 124, 125, 453, 704, 957, 958, 1112, 1113
<code>\glslistdottedwidth</code>	290, 290, 591, 609, 844, 862
<code>\glslistexpandedname</code> ..	287, 287, 861
<code>\glslistgroupheaderfmt</code> ...	287, 287, 288, 289, 608, 609, 861, 862
<code>\glslistinit</code>	287, 288, 861
<code>\glslistnavigationitem</code>	287, 288, 289, 608, 861, 862
<code>\glslocalreset</code> ...	109, 110, 441, 442, 692, 693, 949, 950, 1104, 1105
<code>\glslocalresetall</code>	110, 442, 693, 950, 1105
<code>\glslocalunset</code> ...	109, 111, 124, 442, 443, 463–466, 693, 694, 715–717, 950, 951, 957, 1105, 1112
<code>\glslocalunsetall</code>	111, 443, 694, 951, 1105
<code>\glslocationcstoencap</code> .	53, 915, 1070
<code>\gslongaccessdisplay</code> .	357, 363–366, 373–382, 556, 563–565, 570–578, 809, 817–819, 823–831
<code>\gslongaccesskey</code>	382, 582, 835
<code>\gslongkey</code>	251, 382, 520, 582, 772, 835, 1027, 1183
<code>\gslongpluralaccessdisplay</code> ...	357, 364, 373, 374, 376, 378–380, 556, 564, 570, 571, 573–576, 578, 809, 817, 823, 824, 826–829, 831
<code>\gslongpluralaccesskey</code>	383, 582, 835
<code>\gslongpluralkey</code>	252, 383, 520, 582, 772, 835, 1027, 1183
<code>\gslongtok</code>	255, 256–260, 264, 266, 373, 377, 379, 382, 522–525, 529, 530, 532–540, 542–546, 570, 571, 574, 575, 578–581, 774–777, 780, 782, 784–793, 795–799, 823, 824, 827, 828, 831–835, 1029–1032, 1035, 1036, 1184–1187, 1191, 1192
<code>\gslowercase</code>	4, 888, 1043

<code>\glsLTpenaltycheck</code>	621, 631, 632, 634, 859, 861, 862, 872–875, 884–886, 888, 1042
..... 301, <u>301</u> , 614, 615, 868	
<code>\glsmakefirsttuc</code>	42, 911, 1065
<code>\glsmcols</code>	308, 308, 309–313, 618–621, 872–875
<code>\glsmeasuredepth</code>	<u>108</u> , 949, 1104
<code>\glsmeasureheight</code>	1012, 1021, 1047, 1086, 1167, 1177
.. <u>108</u> , 138, 949, 968, 1104, 1122	
<code>\glsmeasurewidth</code>	86, 428, 678, 934, 1089
..... <u>108</u> , 333–335, 949, 1104	
<code>\glsmfuaddmap</code>	5, 6, 94, 115, 116, 126, 141, 143, 146–160, 162, 164, 166, 168, 242, 253, 254, 340, 342, 343, 888, 889, 939, 940, 953, 958, 969–977, 979–981, 983–986, 1023, 1028, 1043, 1094, 1108, 1113, 1124– 1135, 1137, 1178, 1179, 1183, 1184
<code>\glsmfublocker</code>	5, 6, 94, 142, 144, 146, 148–162, 164, 166, 168, 888, 889, 940, 969–981, 1043, 1094, 1125–1133, 1135–1137
<code>\glsmfuexcl</code>	5, <u>5</u> , 6, 171, 173–179, 181, 888, 889, 988, 1043, 1138–1143
<code>\glsmoveentry</code> <u>104</u> , 440, 690, 947, 1101	
<code>\GLSname</code>	<u>151</u> , 468, 719, 973, 1129
<code>\Glsname</code> 150, <u>150</u> , 468, 719, 973, 1128	
<code>\glsname</code> 150, <u>150</u> , 468, 719, 973, 1128	
<code>\glsnameaccessdisplay</code>	355, 368, 371, 372, 555, 566, 567, 569, 570, 808, 820, 822, 823
<code>\glsnamefont</code>	242, 247, 371, 372, 515, 516, 518, 549, 569, 767, 768, 770, 802, 822, 1022, 1023, 1025, 1178, 1180
<code>\glsnavhypergroupdotarget</code> .	<u>282</u> , <u>282</u>
<code>\glsnavhyperlink</code>	<u>280</u> , 283, 284, 605, 606, 858, 859, 1042, 1043
<code>\glsnavhyperlinkname</code>	280, <u>281</u> , 282, 605, 858, 1042
<code>\glsnavhypertarget</code>	<u>282</u> , 287–289, 309–313, 330, 331, 333, 336, 605, 608, 609, 619– 622, 631, 632, 634, 858, 861, 862, 872–875, 884–886, 888, 1042
<code>\glsnavigation</code>	283, 287–289, 309–313, 330–332, 336, 605, 608, 619– 621, 631, 632, 634, 859, 861, 862, 872–875, 884–886, 888, 1042
<code>\glsnavigationitem</code>	283, <u>283</u>
<code>\glsnextpages</code>	12, 81, 219, <u>239</u> , 389, 425, 504, 514, 638, 675, 756, 766, 893, 931, 1012, 1021, 1047, 1086, 1167, 1177
<code>\glsnoexpandfields</code>	86, 428, 678, 934, 1089
<code>\glsnogroupskipfalse</code>	14, 389, 639, 893, 1048
<code>\glsnoidxdisplayloc</code>	202, 204, <u>235</u> , 496, 497, 512, 748, 749, 764, 1003, 1004, 1019, 1159, 1160, 1175
<code>\glsnoidxdisplayloclisthandler</code>	202, <u>235</u> , 496, 511, 748, 763, 1003, 1019, 1158, 1175
<code>\glsnoidxloclist</code>	201, 232, 234, <u>234</u> , 495, 511, 747, 762, 763, 1002, 1018, 1019, 1158, 1174
<code>\glsnoidxloclisthandler</code>	234, <u>234</u> , 511, 763, 1019, 1174
<code>\glsnoidxnumberlistloophandler</code>	202, <u>203</u> , 496, 497, 748, 1003, 1004, 1159
<code>\glsnoidxprenumberlist</code>	<u>82</u> , 233, 234, 931, 1018, 1086, 1174
<code>\glsnoidxstripaccents</code>	26, <u>26</u> , 396, 397, 646, 647, 900, 901, 1055, 1056
<code>\glsnomakeindexwarning</code>	57, 190, 412, 487, 662, 739, 918, 994, 1072, 1149
<code>\glsnonextpages</code>	81, 219, <u>239</u> , 425, 504, 514, 675, 756, 766, 931, 1012, 1021, 1086, 1167, 1177
<code>\glsnopostdotfalse</code>	13, 389, 639, 893, 1048
<code>\glsnoxindywarning</code>	52, <u>57</u> , 59–61, 63–66, 183, 184, 409, 412–414, 416, 417, 482, 483, 659, 662– 664, 666–668, 734, 735, 915, 918–920, 922, 923, 989, 1070, 1072–1075, 1077, 1078, 1144, 1145
<code>\glsnumberformat</code>	<u>53</u> , 179, 409, 480, 481, 660, 732, 915, 987, 1070, 1142
<code>\glsnumberlistloop</code>	202, <u>203</u> , 496, 748, 1003, 1004, 1159
<code>\glsnumbersgroupname</code>	41, <u>48</u> , 50, 245, 405, 407, 408, 517,

	655, 657, 659, 769, 910, 913, 914, 1024, 1065, 1067, 1069, 1179	\GLSplural .. 149 , 467, 719, 972, 1128
\glsnumlistlastsep 180 , 180 , 202, 481, 496, 733, 747, 987, 1003, 1143, 1158	\Glsplural 148 , 148 , 467, 719, 972, 1128
\glsnumlistparser 180, 199, 481, 494, 733, 745, 987, 1001, 1143, 1156	\glsplural
\glsnumlistsep 180 , 180 , 202, 481, 496, 733, 747, 987, 1003, 1143, 1158 148 , 148 , 467, 719, 972, 1127, 1128
\glsopenbrace ..	63, 182 , 187, 188, 416, 482, 485, 587, 588, 666, 733, 737, 840, 841, 921, 922, 988, 991, 992, 1076, 1144, 1147	\glspluralaccessdisplay
\glsorder ..	32 , 33, 194–196, 198, 230, 401, 491–493, 508, 650, 651, 742–745, 760, 905, 997– 1000, 1015, 1060, 1153–1156, 1171 356 , 359, 360, 367, 555, 557, 558, 560, 561, 566, 808, 810, 811, 813, 814, 819
\glsorg@endtheglossary 6, 6 , 385, 635, 889, 1044	\glspluralsuffix
\glsorg@PrintChanges 6, 385, 635, 889, 1044 48, 48, 101, 102, 256, 259, 260, 373, 374, 378–381, 407, 438, 524, 525, 570, 571, 575, 577, 657, 688, 689, 776, 777, 824, 828–830, 913, 945, 1029, 1031, 1032, 1067, 1099, 1185, 1187, 1188
\glsorg@theglossary 6, 6 , 385, 635, 889, 1044	\glspostdescription
\glspagelistwidth 291 , 293, 296, 297, 299, 300, 302 , 304–307, 314 , 316, 317, 319, 320, 321 , 323–327, 609– 612, 614–618, 622–629, 863, 864, 866–872, 876, 877, 879–883 13 , 50, 286, 288, 289, 291, 303, 314, 321, 322, 329, 330, 332, 334, 335, 389, 409, 590–598, 607–610, 615, 622, 626, 630–634, 639, 659, 844, 846–852, 860–863, 869, 876, 880, 884–887, 893, 914, 1048, 1069
\glspar	51 , 409, 659, 914, 1069	\glspostinline
\glspatchLTOoutput 297 –300, 301 , 613, 614, 866–868 284, 286 , 606, 607, 859, 860
\glspatchtabularx 108 , 441, 692, 949, 1104	\glspostlinkhook
\glspenaltygroupskip 298, 299, 301 , 613, 614, 866–868 125 , 126 , 141–146, 162–169, 364–366, 454, 463–466, 472–476, 564– 566, 705, 715–718, 724–728, 817–819, 958, 969–971, 977– 982, 1113, 1124–1127, 1133–1137
\glspercentchar 89, 183 , 187, 188, 429, 430, 482, 485, 486, 680, 681, 734, 736, 738, 936, 937, 988, 991, 992, 1091, 1144, 1147, 1148	\glsprefixsep
\GLSpl	144 , 465, 717, 970, 1126 341, 341 , 342–345, 602–605, 855–858
\Glspl	116, 143, 143 , 270, 446, 465, 532, 696, 716, 784, 954, 970, 1039, 1108, 1125, 1194	\glsprestandardsort
\glspl ...	115, 142 , 143, 269, 445, 464, 532, 696, 716, 784, 953, 969, 970, 1039, 1108, 1125, 1194 15 , 16, 390, 391, 640, 894, 895, 1049
		\glsquote 44, 182 , 482, 734, 988, 1144
		\glsrefentry 240 , 514, 766, 1022, 1177
		\glsreset
	 108 , 110, 441, 442, 692, 693, 949, 950, 1104, 1105
		\glsresetall . 110 , 442, 693, 950, 1105
		\glsresetcurrcountfalse
	 111, 951, 1106
		\glsresetentrycounter
	 239 , 514, 766, 1021, 1177
		\glsresetentrylist 187, 232, 238, 239 , 239 , 485, 510, 513, 514, 587, 737, 761, 765, 766, 840, 991, 1017, 1021, 1147, 1173, 1177
		\glsresetsubentrycounter
	 239 , 240, 285,

514, 515, 590, 607, 766, 767,
 843, 860, 1021, 1022, 1177, 1178
 \glsrestoreLTOuput ... 301, 614, 868
 \gls sanitizesortfalse
 28, 398, 648, 902, 1057
 \gls sanitizesorttrue
 28, 398, 647, 648, 902, 1056, 1057
 \gls save number list false
 12, 389, 638, 892, 1047
 \gls save writes false
 38, 403, 653, 909, 1063
 \gls see 214, 502, 754, 1010, 1165
 \gls see format . 186, 200, 202, 215,
 484, 495, 496, 502, 587, 736,
 746, 748, 754, 840, 990, 1002,
 1003, 1010, 1146, 1157, 1159, 1165
 \gls see item
 215, 215, 502, 503, 754, 1010, 1166
 \gls see item format
 .. 215, 215, 503, 754, 1010, 1166
 \gls see last sep
 .. 215, 215, 502, 754, 1010, 1166
 \gls see list
 .. 215, 215, 502, 754, 1010, 1165
 \gls see sep
 .. 215, 215, 502, 754, 1010, 1166
 \gls sentence case
 .. 4, 122, 126, 146, 170–172,
 888, 957, 958, 971, 982, 983,
 1043, 1111–1113, 1126, 1138, 1139
 \gls set Alpha Compositor
 52, 409, 659, 915, 1070
 \gls set Compositor
 52, 409, 659, 915, 1070
 \gls set expand field 24, 24, 27–29,
 395, 397, 398, 645, 647, 648,
 899, 901, 902, 1054, 1056, 1057
 \gls set no expand field
 24, 24, 25, 27, 28,
 395, 397, 398, 645, 647, 648,
 899, 901, 902, 1054, 1056, 1057
 \Gls set Quote 98,
 188, 190, 435, 486, 487, 490,
 686, 737, 739, 742, 942, 992,
 994, 996, 1097, 1148, 1149, 1152
 \gls set Suffix F 52, 409, 659, 915, 1070
 \gls set Suffix FF 53, 409, 659, 915, 1070
 \gls set to ctitle 47, 50, 218,
 407, 408, 504, 657, 658, 756,
 912, 914, 1011, 1067, 1069, 1167
 \gls set widest 333, 633, 886
 \Gls set Write Ist Hook
 190, 487, 739, 994, 1149
 \Gls set Xdy Code Page
 66, 417, 667, 668, 923, 1078
 \Gls set Xdy First Letter After Digits
 ... 183, 482, 734, 988, 989, 1144
 \Gls set Xdy Language
 65, 417, 667, 923, 1078
 \Gls set Xdy Location Class Order ..
 64, 416, 666, 922, 1077
 \Gls set Xdy Min Range Length
 ... 184, 483, 734, 735, 989, 1145
 \Gls set Xdy Number Group Order
 183, 482, 483, 734, 989, 1144, 1145
 \Gls set Xdy Styles
 ... 65, 417, 667, 923, 1077, 1078
 \gls short access display ... 172,
 357, 362–365, 373–382, 477,
 556, 562–565, 570–578, 729,
 809, 816–818, 823–831, 983, 1139
 \gls short access key 382, 582, 835
 \gls short accsupp . 355, 355, 554, 807
 \gls short key 251,
 382, 520, 582, 772, 835, 1027, 1183
 \gls short placcsupp 355, 554, 807
 \gls short plural access display ..
 357, 361, 362, 364, 373,
 374, 378–380, 556, 562, 564,
 570, 571, 574–576, 578, 809,
 815, 817, 823, 824, 827–829, 831
 \gls short plural access key
 382, 582, 835
 \gls short plural key 251,
 382, 520, 582, 772, 835, 1027, 1183
 \gls short tok 255,
 256, 382, 522, 532–540, 542–
 546, 578–582, 774, 784–799,
 831–835, 1028, 1029, 1184, 1185
 \gls show accsupp 7, 9,
 386, 387, 636, 637, 890, 891, 1045
 \gls show target 8,
 8, 386, 387, 636, 890, 891, 1045
 \gls show target font
 ... 8, 8, 386, 387, 636, 890, 1045
 \gls show target font text 8, 8, 890, 1045
 \gls show target inner .. 8, 8, 890, 1045
 \gls show target outer
 8, 8, 387, 636, 890, 1045

<code>\glsshowtargetsymbol</code>	624, 626–628, 630–634, 767–769, 843–852, 860–865, 869–871, 876–878, 880–882, 884–887, 948, 949, 1022, 1103, 1178
<code>\glssortnumberfmt</code> 16, 17, 18, 391, 392, 640, 641, 895, 896, 1049, 1050	
<code>\glsspace</code>	252, 252, 521, 772, 785, 1027, 1183
<code>\glstepentry</code> . 239, 240, 514, 515, 766, 767, 1021, 1022, 1177, 1178	
<code>\glstepsubentry</code> 239, 240, 514, 515, 766, 767, 1021, 1022, 1177, 1178	
<code>\glssubentrycounterfalse</code>	15, 390, 640, 894, 1049
<code>\glssubentrycounterlabel</code> .. 240, 240, 514, 515, 766, 767, 1022, 1178	
<code>\glssubentryitem</code>	240, 285, 288–291, 293, 295, 303, 304, 306, 314, 316, 318, 322, 323, 325, 329, 330, 332, 334, 515, 590–599, 607–611, 615–617, 622–624, 626–628, 630–633, 767, 843–852, 860–865, 869–871, 876–878, 880–882, 884, 885, 887, 1022, 1178
<code>\GLSsymbol</code> .. 154, 469, 721, 974, 1130	
<code>\GLSsymbol</code> 153, 153, 469, 720, 974, 1130	
<code>\glssymbol</code>	153, 153, 469, 720, 974, 1129, 1130
<code>\glssymbolaccessdisplay</code> .. 356, 369, 372, 555, 558–560, 567, 569, 570, 808, 812, 813, 820, 823	
<code>\glssymbolnav</code> ... 284, 606, 859, 1043	
<code>\GLSsymbolplural</code>	155, 469, 721, 975, 1130
<code>\GLSsymbolplural</code>	154, 154, 469, 721, 974, 1130
<code>\glssymbolplural</code>	154, 154, 469, 721, 974, 1130
<code>\glssymbolpluralaccessdisplay</code> .	356, 369, 555, 557, 558, 567, 808, 810, 811, 821
<code>\glssymbolsgroupname</code>	40, 47, 50, 245, 404, 407, 408, 517, 655, 657, 659, 769, 910, 913, 914, 1024, 1064, 1067, 1069, 1179
<code>\glstarget</code>	107, 108, 241, 286–291, 293–295, 303, 304, 306, 314, 316, 318, 321–323, 325, 329, 330, 332, 334, 335, 515–517, 590–599, 607–611, 615–617, 622–624, 626–628, 630–634, 767–769, 843–852, 860–865, 869–871, 876–878, 880–882, 884–887, 948, 949, 1022, 1103, 1178
<code>\glstexorpdfstring</code>	170, 170, 171, 175, 208, 982, 985, 1007, 1138, 1140, 1162
<code>\GLStext</code> 146, 466, 718, 971, 1127	
<code>\Glstext</code> 147, 147, 466, 718, 971, 1127	
<code>\glstext</code> 146, 147, 466, 718, 971, 1127	
<code>\glstextaccessdisplay</code> . 356, 360, 361, 366, 555, 558, 559, 561, 562, 566, 808, 812, 814, 815, 819	
<code>\glstextformat</code> ... 117, 125, 127, 446, 454, 455, 697, 705, 706, 954, 958, 959, 1109, 1113, 1114	
<code>\glstextup</code>	48, 251, 380, 407, 520, 577, 657, 772, 830, 913, 1027, 1067, 1183
<code>\glstildechar</code>	59, 183, 187, 188, 413, 482, 485, 663, 734, 736, 737, 919, 988, 991, 992, 1074, 1144, 1147
<code>\glstranslatefalse</code>	30, 31, 399, 400, 649, 903, 904, 1058, 1059
<code>\glstranslatetrue</code> .. 30, 31, 399, 400, 649, 650, 903, 904, 1058, 1059	
<code>\glstreechildpredesc</code>	328, 329, 330, 630, 631, 883–885
<code>\glstreegroupheaderfmt</code>	308–313, 327, 329–333, 336, 619–622, 629–632, 634, 872–875, 883–886, 888
<code>\glstreeindent</code>	330, 331, 332, 334, 335, 594–596, 631–634, 848, 849, 885, 887
<code>\glstreeitem</code> .. 308, 309, 328, 328, 618, 619, 629, 630, 872, 873, 883	
<code>\glstreenamebox</code>	333, 334, 335, 633, 634, 886, 887
<code>\glstreenamefmt</code> .. 327, 327, 329, 330, 332–335, 629–634, 883–887	
<code>\glstreenavigationfmt</code>	309–313, 327, 330–332, 336, 619–621, 629, 631, 632, 634, 872–875, 883–886, 888
<code>\glstreepredesc</code>	328, 329, 330, 332, 630–632, 883–885
<code>\glstreesubitem</code>	308, 328, 328, 618, 630, 872, 883

<code>\glstreesubsubitem</code>	<code>\glsuseriaccessdisplay</code>
308, 328, <u>328</u> , 618, 630, 872, 883 357, 369, 556, 568, 809, 821
<code>\glstype</code>	<code>\GLSuserii</code>
126, 127, 141–145,	157, 470, 722, 975, 1131
162–169, 364–366, 454, 463–	<code>\Glsuserii</code> 156, <u>156</u> , 470, 722, 975, 1131
466, 472–476, 564, 565, 705,	<code>\glsuserii</code> 156, <u>156</u> , 470, 721, 975, 1131
715–717, 723–728, 817–819,	<code>\glsuseriiaccessdisplay</code>
958, 959, 969–971, 977–982, 357, 370, 556, 568, 809, 821
1113, 1114, 1124–1126, 1133–1137	<code>\GLSuseriii</code>
<code>\glsucmarkfalse</code> 14, 389, 639, 893, 1048	158, 470, 722, 976, 1131
<code>\glsucmarktrue</code> 14, 389, 639, 893, 1048	<code>\Glsuseriii</code>
<code>\glsunexpandedfieldvalue</code> 157, <u>157</u> , 470, 722, 976, 1131
. . . . <u>96</u> , 287, 685, 861, 941, 1096	<code>\glsuseriii</code>
<code>\glsunset</code>	157, <u>157</u> , 470, 722, 975, 976, 1131
109,	<code>\glsuseriiiaccessdisplay</code>
111–113, 124, 441–444, 463– 358, 370, 556, 568, 809, 821
466, 692–695, 715–718, 949,	<code>\GLSuseriv</code>
950, 952, 957, 1104–1107, 1112	159, 471, 722, 976, 1132
<code>\glsunsetall</code>	<code>\Glsuseriv</code> 158, <u>158</u> , 471, 722, 976, 1132
111, 443, 693, 950, 1105	<code>\glsuseriv</code>
<code>\glsupacrpluralsuffix</code>	158, <u>158</u> , 470, 722, 976, 1131, 1132
. . . . 48, 261, 267, 407, 526, 531,	<code>\glsuserivaccessdisplay</code>
535, 538, 540, 543, 657, 778,	358, 370, 556, 568, 809, 821, 822
783, 788, 791, 793, 796, 913,	<code>\GLSuseriv</code>
1032, 1033, 1038, 1067, 1188, 1193	160, 471, 723, 977, 1132
<code>\glsupercase</code>	<code>\Glsuseriv</code> 159, <u>159</u> , 471, 723, 976, 1132
. . . . 4, 55, 93, 118–121, 147–	<code>\glsuseriv</code> 159, <u>159</u> , 471, 722, 976, 1132
161, 163, 165, 167, 169, 256,	<code>\glsuserivaccessdisplay</code>
257, 264–266, 344, 345, 360–	358, 371, 556, 568, 569, 810, 822
363, 365–371, 376–378, 380,	<code>\GLSuserivi</code>
888, 916, 939, 954–956, 971–	161, 471, 723, 977, 1133
980, 982, 1029, 1035–1037,	<code>\Glsuserivi</code> 160, <u>160</u> , 471, 723, 977, 1132
1043, 1071, 1094, 1109–1111,	<code>\glsuserivi</code> 160, <u>160</u> , 471, 723, 977, 1132
1127–1137, 1185, 1190, 1191, 1193	<code>\glsuseriviaccessdisplay</code>
<code>\GlsUseAcrEntryDispStyle</code> 358, 371, 556, 569, 810, 822
. 257, <u>258</u> ,	<code>\GlsWarnAddProtectedPageFmt</code>
260–263, 265, 267, 268, 374– 208, <u>208</u> , 1006, 1162
376, 378, 380, 381, 523–529,	<code>\glswrallowprimitivemodsfalse</code>
531, 571, 572, 575–577, 775– 209, 1007, 1163
779, 781, 783, 824–826, 828,	<code>\glswrallowprimitivemodstrue</code>
830, 1030, 1032–1034, 1036– 500, 752
1038, 1186, 1188–1190, 1192–1194	<code>\glswrglossdisableanchorcmds</code>
<code>\GlsUseAcrStyleDefs</code> 208, 213, 248,
. . . . 257, <u>258</u> , 260–263, 265,	1007, 1008, 1026, 1162, 1164, 1181
267, 268, 374–376, 378, 380–	<code>\glswrglossdisablelocationcmds</code>
382, 523, 524, 526–529, 531, 204,
532, 571, 572, 575–577, 775–	207, <u>207</u> , 1004, 1006, 1160, 1161
781, 783, 824–826, 828, 830,	<code>\glswrglosslocationtarget</code>
831, 1030, 1032–1034, 1036–	209, 248, 1007, 1026, 1162, 1181
1038, 1186, 1188–1190, 1192–1194	<code>\glswrglosslocationtextfmt</code>
<code>\GLSuseri</code>	209, 249, 1007, 1026, 1162, 1181
156, 470, 721, 975, 1131	<code>\glswrite</code>
<code>\Glsuseri</code> 155, <u>155</u> , 470, 721, 975, 1130	184–190, 198,
<code>\glsuseri</code> 155, <u>155</u> , 469, 721, 975, 1130	200, 205, 483–487, 493, 497,
	498, 585–589, 735–739, 745,

749, 750, 839–842, 989–993,
1000, 1005, 1145–1149, 1156, 1160
`\glswritedefhook` . . . 89, 97, 430,
435, 680, 686, 936, 942, 1091, 1097
`\glswriteentry` 206,
206, 498, 750, 1005, 1006, 1161
`\glswritefiles`
. 38, 204, 403, 497, 653,
749, 908, 909, 1004, 1063, 1160
`\glsxindyfalse` 33, 401, 651, 905, 1060
`\glsxindytrue`
. 34, 401, 651, 905, 906, 1060
`\group` 223, 225

H

`\H` 26, 397, 646, 901, 1055
`\hangindent` 312, 313,
328, 330, 332, 334–336, 594–
596, 621, 622, 629, 631–634,
847–849, 875, 883–885, 887, 888
`\hbox` 108, 290, 441,
591, 609, 692, 844, 862, 949, 1104
`\hfill` 290, 591, 609, 844, 862
`\hline` 292–296, 303–305,
307, 315–317, 319, 320, 322–
327, 610–612, 616–618, 622–
629, 863–865, 869–872, 876–883
`\hsize` 290, 291, 302,
314, 321, 609, 615, 622, 626,
862, 863, 869, 875, 876, 879, 880
`\hspace` 328, 630, 883
`\hss` 290, 591, 609, 844, 862
`\ht` 301, 614, 868
`\hyperbf` . . . 249, 520, 772, 1026, 1182
`\hyperdef` 46, 406, 656, 912, 1067
`\hyperemph` . 249, 520, 772, 1026, 1182
`hyperfirst` (option) 31
`\hyperit` . . . 249, 520, 772, 1026, 1182
`\hyperlink` 123, 139, 249,
453, 462, 519, 704, 714, 771,
957, 968, 1026, 1112, 1123, 1181
`\hypermd` . . . 249, 520, 772, 1026, 1182
`\hyperm` . . . 249, 520, 771, 1026, 1182
`\hypersc` . . . 249, 520, 772, 1026, 1182
`\hypersf` . . . 249, 520, 772, 1026, 1182
`\hypersl` . . . 249, 520, 772, 1026, 1182
`\hypertarget` . 138, 462, 714, 968, 1122
`\hypertt` . . . 249, 520, 772, 1026, 1182
`\hyperup` . . . 249, 520, 772, 1026, 1182

I

`\IeC` 26, 396, 646, 900, 1055
`\if` 129, 130,
212, 456, 501, 585, 707, 753,
838, 960, 961, 1008, 1115, 1164
`\if@endfor` 605, 858, 1042
`\if@endpe` 220, 1168
`\if@filesw` 34, 906, 1060
`\if@gl@debug` 7, 23, 206,
386, 394, 498, 635, 644, 750,
889, 898, 1005, 1044, 1053, 1161
`\if@gl@docloaded`
. 6, 6, 19, 385, 392, 404,
635, 642, 654, 889, 896, 1044, 1051
`\if@gl@sisacronymlist` 22, 126, 394,
454, 643, 705, 898, 958, 1052, 1113
`\if@openright` . 56, 411, 661, 917, 1072
`\if@org@KV@gl@link@local`
. 123, 124, 957, 1112
`\ifbool` . 38, 39, 68, 393, 400, 403,
419, 446–448, 642, 650, 653,
669, 697–699, 909, 924, 1063, 1079
`\ifboolexpr` . 49, 74, 407, 422, 517,
657, 672, 769, 913, 928, 1068, 1083
`\ifcase` 7,
9, 10, 30, 37, 40, 41, 81, 237,
329, 386–388, 399, 404, 425,
513, 594, 630, 635, 637, 638,
649, 654, 675, 764, 847, 884,
889, 891, 892, 903, 907, 909,
911, 931, 1020, 1044, 1046,
1058, 1062, 1064, 1065, 1086, 1176
`\ifcsdef` 30, 50, 56, 84,
85, 92–97, 107, 206, 227, 231,
258, 354, 399, 408, 411, 427,
428, 432–435, 452, 498, 506,
508, 509, 518, 523, 554, 649,
658, 661, 677, 678, 682–686,
703, 750, 758, 760, 761, 770,
775, 807, 903, 914, 917, 933,
934, 938–942, 948, 1005, 1013,
1015–1017, 1030, 1042, 1058,
1069, 1072, 1088, 1093–1096,
1103, 1161, 1169, 1171, 1172, 1186
`\ifcsempy` 71, 340, 341,
420, 601, 670, 854, 855, 926, 1080
`\ifcsequal` . . . 71, 420, 670, 926, 1080
`\ifcsname` 96, 685, 941, 1096
`\ifcsstrequal` . 97, 435, 686, 942, 1096
`\ifcsstring` . . . 96, 434, 685, 941, 1096

<code>\ifcsundef</code>	780, 786, 789, 791, 794, 810, 813, 815, 826, 898, 907, 917, 924, 926, 927, 929, 930, 954, 955, 999–1001, 1017, 1018, 1029, 1030, 1034, 1039, 1053, 1061, 1071, 1079, 1081, 1082, 1084, 1109, 1110, 1155, 1156, 1173, 1174, 1184, 1185, 1190, 1195
10, 17, 33, 46, 49, 53–56, 68, 76, 78, 80, 99, 102, 112, 123, 128, 129, 139, 204, 216, 227, 235, 236, 241, 244–247, 250, 257, 355–358, 388, 391, 401, 406, 407, 409–411, 418, 419, 422–424, 436, 438, 443, 453, 455, 462, 463, 491, 492, 497, 503, 505, 506, 512, 515, 517, 518, 520, 523, 555–557, 589, 605, 637, 641, 651, 656, 658, 660, 661, 669, 673, 675, 687, 689, 694, 704, 706, 714, 743, 749, 755, 757, 758, 764, 767, 769, 770, 772, 775, 808–810, 843, 859, 892, 895, 905, 912, 913, 915–917, 924, 929, 930, 943, 945, 951, 957, 959, 960, 968, 1004, 1010, 1014, 1019, 1020, 1022–1026, 1030, 1042, 1046, 1050, 1060, 1067, 1068, 1070–1072, 1079, 1083–1085, 1098, 1100, 1106, 1112, 1114, 1115, 1123, 1124, 1160, 1166, 1169, 1175, 1178–1180, 1182, 1185	<code>\ifdefequal</code>
<code>\ifcsvoid</code>	34, 35, 70–73, 84–86, 90, 100, 104, 234, 355, 401, 402, 419–421, 427, 428, 430, 437, 440, 511, 555, 651, 652, 670, 671, 677, 678, 681, 688, 691, 763, 808, 906, 925– 927, 933, 934, 937, 944, 947, 1018, 1060, 1061, 1080–1082, 1088, 1089, 1092, 1098, 1101, 1174
<code>\ifdef</code> 5, 43, 72, 73, 82, 88, 89, 108, 123, 170, 201, 202, 208, 251, 287, 328, 341, 354, 421, 426, 429, 441, 453, 476, 478, 495, 496, 520, 554, 602, 629, 630, 671, 676, 680, 692, 704, 728, 730, 747, 748, 772, 807, 855, 861, 883, 888, 911, 927, 932, 936, 949, 957, 982, 1002, 1003, 1007, 1027, 1043, 1066, 1081, 1082, 1086, 1090, 1091, 1104, 1112, 1138, 1158, 1159, 1162, 1183	<code>\ifdefstrequal</code> 96, 434, 685, 941, 1096
<code>\ifdefempty</code>	<code>\ifdefstring</code>
22, 35, 56, 67, 71–73, 77, 78, 117, 119, 197, 199, 232, 234, 256, 257, 263, 272, 359, 361, 376, 394, 402, 411, 418, 420, 421, 423, 446, 449, 450, 493, 494, 510, 511, 522, 523, 528, 533, 537, 539, 541, 557, 560, 562, 573, 644, 652, 661, 668, 670, 671, 673, 674, 697, 700, 701, 745, 746, 762, 763, 774, 775,	... 13, 49, 74, 194–196, 199, 229, 230, 233–235, 346, 389, 407, 422, 491–493, 508–512, 549, 639, 657, 672, 742–744, 746, 759, 760, 762, 763, 802, 893, 913, 928, 997–999, 1001, 1015–1017, 1019, 1048, 1068, 1083, 1153–1155, 1157, 1171–1175
	<code>\ifdefvoid</code> ... 25, 26, 34, 48, 103, 195, 196, 233, 234, 396, 439, 510, 511, 645, 646, 690, 762, 763, 900, 905, 913, 946, 997, 998, 1018, 1054, 1055, 1060, 1068, 1101, 1153, 1154, 1173, 1174
	<code>\ifdim</code>
	260, 301, 333, 525, 614, 633, 777, 868, 886, 1032, 1187
	<code>\iffalse</code>
	103, 110, 439, 442, 690, 693, 946, 950, 1101, 1105
	<code>\IfFileExists</code> ... 12, 29, 30, 196, 197, 220, 389, 399, 491, 492, 505, 639, 648, 649, 743, 744, 756, 893, 903, 998–1000, 1012, 1047, 1057, 1058, 1154, 1155, 1168
	<code>\ifglossaryexists</code>
	54, 65, 67, 70, 194–196, 410, 417– 419, 490–492, 660, 667–669, 742–744, 915, 923–925, 997– 999, 1070, 1078–1080, 1152–1154
	<code>\ifglS@sanitize@description</code> 27, 397, 647, 901, 1056

<code>\ifgls@sanitize@name</code>	<code>\ifglshasfield</code> 72 , 421 , 671 , 926 , 1081
..... 27 , 397 , 647 , 901 , 1056	<code>\ifglshaslong</code>
<code>\ifgls@sanitize@symbol</code> 71 , 114 – 116 , 171 , 258 –
..... 28 , 397 , 647 , 902 , 1056	260 , 263 , 266 , 373 , 376 , 379 ,
<code>\ifgls@xindy@glslnumbers</code>	420 , 445 , 446 , 477 , 524 , 525 ,
..... 66 , 418 , 668 , 923 , 1078	528 , 530 , 539 , 570 , 571 , 573 ,
<code>\ifglsacrdescription</code>	575 , 670 , 696 , 697 , 728 , 776 ,
..... 32 , 544 , 798 , 905 , 1059	777 , 780 , 781 , 792 , 823 , 824 ,
<code>\ifglsacrdua</code> .. 32 , 535 , 540 , 543 –	826 , 828 , 926 , 953 , 954 , 983 ,
545 , 788 , 793 , 796 , 798 , 905 , 1060	1030 – 1032 , 1035 , 1036 , 1081 ,
<code>\ifglsacrfootnote</code> 32 , 126 , 454 , 544 ,	1108 , 1138 , 1186 , 1187 , 1190 , 1192
545 , 705 , 798 , 904 , 958 , 1059 , 1113	<code>\ifglshasparent</code>
<code>\ifglsacronym</code>	70 ,
..... 20 , 392 , 393 , 642 , 897 , 1051 , 1052	222 , 227 , 233 , 333 , 420 , 506 ,
<code>\ifglsacrshortcuts</code>	510 , 632 , 670 , 758 , 762 , 886 ,
..... 45 , 270 , 405 , 532 ,	926 , 1014 , 1018 , 1080 , 1169 , 1173
656 , 784 , 911 , 1039 , 1066 , 1194	<code>\ifglshasprefix</code>
<code>\ifglsacrsmalldcaps</code>	340 ,
32 ,	341 , 343 , 344 , 601 – 604 , 854 – 857
535 , 536 , 538 , 540 , 543 , 787 –	<code>\ifglshasprefixfirst</code>
789 , 791 , 793 , 795 , 796 , 905 , 1059	340 ,
<code>\ifglsacrsmaller</code> 32 , 535 , 536 , 539 ,	341 , 343 , 344 , 601 – 604 , 855 – 857
540 , 787 – 789 , 791 , 793 , 905 , 1059	<code>\ifglshasprefixfirstplural</code>
<code>\ifglsautomake</code>	341 , 342 , 344 ,
... 36 , 37 , 43 , 199 , 402 , 403 ,	345 , 601 , 602 , 604 , 605 , 855 – 858
494 , 652 , 653 , 745 , 907 , 908 ,	<code>\ifglshasprefixplural</code> . 340 , 342 ,
911 , 1001 , 1062 , 1063 , 1066 , 1156	343 , 345 , 601 – 604 , 854 , 856 – 858
<code>\ifglsdescsuppressed</code>	<code>\ifglshashshort</code>
71 ,	71 , 215 ,
285 , 420 , 606 , 670 , 860 , 926 , 1080	420 , 671 , 926 , 1010 , 1081 , 1166
<code>\ifglsentrycounter</code>	<code>\ifglshassymbol</code>
... 14 , 15 , 45 , 239 , 240 , 390 ,	71 , 329 ,
406 , 514 , 515 , 639 , 640 , 656 ,	330 , 332 , 334 , 335 , 420 , 534 ,
766 , 767 , 893 , 894 , 911 , 1021 ,	537 , 541 , 630 – 634 , 670 , 786 ,
1022 , 1048 , 1049 , 1066 , 1177 , 1178	789 , 794 , 884 , 885 , 887 , 926 , 1081
<code>\ifglsentryexists</code> ... 68 , 69 , 87 ,	<code>\ifglshyperfirst</code>
88 , 98 , 100 , 418 , 419 , 429 , 435 , 126 , 454 , 705 , 958 , 1113
437 , 669 , 679 , 680 , 686 , 688 ,	<code>\ifglsindexonlyfirst</code>
924 , 925 , 935 , 936 , 942 , 944 , 207 , 498 , 750 , 1006 , 1161
1079 , 1080 , 1089 – 1091 , 1097 , 1099	<code>\ifglsnogroupskip</code>
<code>\ifglsesclocations</code>	288 ,
.... 130 , 207 , 208 , 499 , 751 ,	292 , 293 , 295 , 298 , 299 , 303 ,
961 , 1006 , 1007 , 1116 , 1161 , 1162	304 , 306 , 314 , 316 , 318 , 322 ,
<code>\ifglsfieldcseq</code> 97 , 435 , 686 , 942 , 1096	324 , 326 , 329 , 331 , 332 , 335 ,
<code>\ifglsfielddefeq</code>	608 , 610 , 611 , 613 , 616 , 617 ,
..... 96 , 434 , 685 , 941 , 1096	622 – 624 , 626 – 628 , 630 – 632 ,
<code>\ifglsfieldeq</code> . 96 , 434 , 685 , 941 , 1096	634 , 861 , 863 – 867 , 869 – 871 ,
<code>\ifglsfieldvoid</code>	876 – 878 , 880 – 882 , 884 – 886 , 888
72 , 926 , 1081	<code>\ifglsnonumberlist</code>
<code>\ifglshaschildren</code> . 70 , 285 , 419 , 238 , 513 , 765 , 1021 , 1176
590 , 607 , 669 , 843 , 860 , 925 , 1080	<code>\ifglsnopostdot</code> 13 , 389 , 639 , 893 , 1048
<code>\ifglshasdesc</code>	<code>\ifglsnumberline</code>
71 , 57 , 412 , 662 , 917 , 1072
285 , 420 , 607 , 670 , 860 , 926 , 1080	<code>\ifglsresetcurrcount</code>
 111 , 112 , 951 , 1106

<code>\ifglssanitizesort</code>	1011, 1084, 1098, 1101, 1161, 1166
.. 25, 28, 395, 398, 645, 647,	
648, 899, 902, 1054, 1056, 1057	<code>\ifin@</code>
<code>\ifglssavenumberlist</code> 84, 199, 216,	42, 405, 655, 911, 1065
427, 494, 503, 677, 745, 754,	<code>\ifinlistcs</code> ... 231, 236, 509, 512,
933, 1000, 1010, 1087, 1156, 1166	761, 764, 1016, 1019, 1172, 1175
<code>\ifglssavewrites</code> .. 38, 194, 206,	<code>\ifinner</code>
403, 490, 498, 653, 742, 750,	8, 386, 636, 890, 1045
908, 997, 1005, 1063, 1152, 1161	<code>\ifKV@glslink@hyper</code> ... 126–128,
<code>\ifglssubentrycounter</code> ... 15, 45,	454, 455, 705, 706, 959, 1113, 1114
239, 240, 390, 406, 514, 515,	<code>\ifKV@glslink@local</code>
640, 656, 766, 767, 894, 911,	124, 463–466, 715–717, 957, 1112
1021, 1022, 1049, 1066, 1177, 1178	<code>\ifmeasuring@</code> 106, 441, 692, 948, 1102
<code>\ifglstoc</code> 57, 412, 662, 917, 1072	<code>\ifmmode</code>
<code>\ifglstranslate</code> 29, 48, 399,	8, 386, 636, 890, 1045
407, 648, 657, 903, 913, 1057, 1067	<code>\ifnum</code> . 16, 112, 113, 197, 233, 301,
<code>\ifglsucmark</code>	330, 332, 334, 335, 391, 402,
.... 55, 410, 660, 661, 916, 1071	444, 493, 510, 594–596, 614,
<code>\ifglused</code>	631–633, 640, 641, 653, 694,
. 68, 114, 117–119, 126, 182,	695, 744, 762, 848, 849, 868,
207, 341–345, 359–361, 419,	885, 887, 895, 951, 952, 999,
445–450, 454, 482, 498, 534,	1017, 1049, 1106, 1107, 1155, 1173
537, 539, 541, 557–562, 602–	<code>\ifstrempty</code>
604, 669, 695, 697–701, 705,	9,
733, 750, 786, 789, 792, 794,	81, 387, 590, 637, 843, 891, 1045
810–815, 855–858, 924, 952,	<code>\ifstrequal</code>
954, 955, 958, 988, 1006, 1079,	517, 769
1107, 1109, 1110, 1113, 1144, 1161	<code>\ifthenelse</code>
<code>\ifglswrallowprimitivemods</code>	29,
.. 209, 211, 500, 752, 1007, 1163	46, 56, 128, 198, 398, 406, 411,
<code>\ifglxsindy</code>	455, 493, 517, 545, 605, 648,
33, 51, 52, 58–61, 63–66, 77,	656, 661, 706, 745, 769, 798,
104, 105, 131, 183, 184, 190,	858, 902, 912, 917, 959, 1000,
194, 195, 197, 212, 214, 220,	1042, 1057, 1066, 1072, 1114, 1156
401, 409, 412–418, 423, 440,	<code>\IfTrackedLanguage</code>
457, 482, 483, 487, 490, 491, 190, 487, 739, 994, 1149
501, 502, 505, 584, 585, 651,	<code>\IfTrackedLanguageFileExists</code> ..
659, 662–668, 673, 691, 708, 49, 337, 338, 408,
734, 735, 739, 742, 743, 752,	583, 600, 658, 836, 853, 914, 1068
754, 757, 837–839, 894, 905,	<code>\iftrue</code>
915, 918–923, 929, 947, 961,	103, 109, 439,
988, 989, 994, 997, 998, 1000,	442, 690, 693, 946, 950, 1101, 1105
1008, 1009, 1012, 1049, 1060,	<code>\ifundef</code>
1069, 1070, 1072–1078, 1084,	14, 15, 75,
1101, 1102, 1116, 1144, 1145,	87, 99, 184, 188, 198, 390, 422,
1149, 1153–1155, 1164, 1165, 1168	429, 436, 483, 486, 493, 639,
<code>\ifignoredglossary</code>	640, 672, 679, 687, 735, 738,
78,	745, 893, 894, 928, 935, 943,
100, 103, 206, 216, 424, 436,	989, 992, 1000, 1048, 1049,
439, 498, 503, 674, 687, 690,	1083, 1090, 1098, 1145, 1148, 1156
750, 755, 930, 943, 946, 1005,	<code>\ifvmode</code> 181, 481, 733, 988, 1143
	<code>\ifvoid</code>
	301, 615, 868
	<code>\ifx</code>
	14, 16, 18, 21, 35,
	41, 42, 46, 58, 59, 61, 63, 66,
	67, 74, 100–103, 105–107, 128,
	129, 132–137, 172, 184, 187,
	189–193, 204, 208, 209, 211–
	215, 218, 245–248, 272, 274,
	346, 355, 390–394, 405, 406,

759, 762, 763, 767, 769, 886, 905, 907, 925–927, 936, 938, 941, 944, 945, 982, 983, 1002, 1003, 1013–1015, 1018, 1022– 1024, 1042, 1060, 1062, 1080– 1082, 1091–1093, 1096, 1099, 1100, 1137, 1138, 1158, 1159, 1169, 1170, 1173, 1174, 1178, 1179	<code>\longnewglossaryentry</code>
<code>list</code> (style) 97, 98, 435, 686, 942, 1097
<code>\listcsadd</code>	<code>longplural</code> (key) 83
<code>\listcseadd</code>	<code>longpluralaccess</code> (key) 348
<code>\listcsgadd</code>	<code>\longprovideglossaryentry</code>
. 235, 236, 512, 764, 1019, 1175 98, 435, 686, 942, 1097
<code>\listcsxadd</code>	<code>longragged</code> (style) 302
<code>listdotted</code> (style)	<code>longragged-booktabs</code> (style) 299
<code>listead</code> 231, 509, 761, 1016, 1172	<code>longragged3col</code> (style) 304
<code>listgroup</code> (style) 288	<code>longragged3col-booktabs</code> (style) 300
<code>listhypergroup</code> (style) 288	<code>longragged3colborder</code> (style) 305
<code>\listxadd</code> 87, 679, 935, 1090	<code>longragged3colheader</code> (style) 305
<code>\loadglsentries</code>	<code>longragged3colheaderborder</code>
. 117, 117, 446, 697, 954, 1108, 1109	(style) 305
<code>locales</code> (option) 31	<code>longraggedborder</code> (style) 303
<code>\long</code> 97, 249, 435, 520,	<code>longraggedheader</code> (style) 303
686, 771, 942, 1026, 1097, 1182	<code>longraggedheaderborder</code> (style) 304
<code>long</code> (key) 83	<code>\LT@end@pen</code> 301, 614, 868
<code>long</code> (style) 291	<code>\LT@err</code> 301, 614, 868
<code>long-booktabs</code> (style) 297	<code>\LT@foot</code> 301, 614, 615, 868
<code>long-sc-short</code> (acrstyle) 260	<code>\LT@head</code> 301, 614, 615, 868
<code>long-sc-short-desc</code> (acrstyle) 262, 374	<code>\LT@lastfoot</code> 301, 614, 615, 868
<code>long-short</code> (acrstyle) 258, 373	<code>\LT@output</code> 301, 614, 868
<code>long-short-desc</code> (acrstyle) 261, 374	
<code>long-sm-short</code> (acrstyle) 261	
<code>long-sm-short-desc</code> (acrstyle) 262, 375	
<code>long-sp-short</code> (acrstyle) 259	
<code>long-sp-short-desc</code> (acrstyle) 262	
<code>long3col</code> (style) 292	
<code>long3col-booktabs</code> (style) 298	
<code>long3colborder</code> (style) 293	
<code>long3colheader</code> (style) 294	
<code>long3colheaderborder</code> (style) 294	
<code>long4col</code> (style) 294	
<code>long4col-booktabs</code> (style) 298	
<code>long4colborder</code> (style) 295	
<code>long4colheader</code> (style) 295	
<code>long4colheaderborder</code> (style) 295	
<code>longaccess</code> (key) 348	
<code>longborder</code> (style) 292	
<code>longheader</code> (style) 292	
<code>longheaderborder</code> (style) 292	

M	
<code>\makeatletter</code>	88, 220, 429, 505,
	680, 756, 936, 1012, 1090, 1168
<code>\makebox</code>	290,
	333–335, 591, 595, 596, 609,
	633, 634, 844, 849, 862, 886, 887
<code>\makefirststuc</code>	4, 888, 1043
<code>\makeglossaries</code>	9,
	34–36, 40, 47, 81, 194, 197,
	199, 201, 203, 221, 387, 401–
	404, 407, 425, 490, 493–495,
	497, 505, 637, 651–654, 657,
	675, 742, 745–747, 749, 757,
	891, 906, 907, 909, 910, 912,
	931, 997, 1000–1002, 1004,
	1013, 1046, 1060–1062, 1064,
	1067, 1086, 1153, 1155–1159, 1168
<code>\makeglossary</code>	39, 39, 40,
	403, 404, 653, 654, 909, 910, 1064
<code>makeindex</code> (option)	33
<code>\makenoidxglossaries</code>	
 9, 34, 35, 81, 199, 200,
	203, 387, 401, 402, 425, 494–
	497, 637, 651, 652, 675, 745–
	749, 891, 906, 931, 1001–1004,
	1046, 1060, 1061, 1086, 1156–1159

`\MakeTextUppercase` 5, 385, 635, 889, 1043
`\MakeUppercase` 558, 559, 565, 811–813, 818, 819
`\marginpar` ... 8, 387, 636, 890, 1045
`\markboth` ... 55, 410, 660, 916, 1071
`\mbox` 181, 289, 312, 313, 334, 481, 591, 608, 621, 633, 733, 844, 862, 874, 875, 887, 988, 1143
`mcolalmtree` (style) [312](#)
`mcolalmtreegroup` (style) [312](#)
`mcolalmtreehypergroup` (style) ... [312](#)
`mcolalmtreeespannav` (style) [313](#)
`mcolindex` (style) [308](#)
`mcolindexgroup` (style) [308](#)
`mcolindexhypergroup` (style) [308](#)
`mcolindexspannav` (style) [309](#)
`mcoltree` (style) [309](#)
`mcoltreegroup` (style) [310](#)
`mcoltreehypergroup` (style) [310](#)
`mcoltreename` (style) [310](#)
`mcoltreenamegroup` (style) [311](#)
`mcoltreenamehypergroup` (style) [311](#)
`mcoltreenameespannav` (style) ... [311](#)
`mcoltreeespannav` (style) [310](#)
`\memUchead` ... 55, 410, 660, 916, 1071
`\MessageBreak` 23, 39, 75, 218, 337, 338, 346, 395, 403, 404, 422, 504, 548, 583, 600, 644, 653, 654, 672, 756, 802, 836, 837, 853, 899, 909, 928, 1011, 1053, 1064, 1083, 1167
`mfirstuc` (option) [41](#)
`\mfirstucMakeUppercase` 5, 385, 410, 433, 447–451, 466–476, 523, 528–530, 537, 541, 561–563, 566–569, 573, 574, 576, 604, 605, 635, 660, 661, 683, 698–702, 718–726, 728, 774, 775, 780–782, 790, 794, 814–816, 819–822, 826, 827, 829, 857, 858, 889, 1043
`\MFUaddmap` 5, 888, 1043
`\MFUblocker` 5, 888, 1043
`\MFUexcl` 5, 888, 1043
`\MFUsentencecase` 5, [5](#), 42, 170, 171, 175, 888, 889, 911, 982, 985, 1043, 1065, 1138, 1140
`\midrule` 298, 299, 613, 866, 867
`\month` 184, 188, 483, 486, 585, 588, 735, 738, 839, 841, 989, 992, 1145, 1148
`\msg` 222

N

`\n` 189, 486, 588, 589, 738, 841, 842, 992, 993, 1148
`name` (key) [78](#)
`\NeedsTeXFormat` 4, 280, 284, 286, 290, 297, 302, 307, 313, 320, 327, 336–338, 345, 384, 385, 548, 583, 584, 589, 600, 635, 801, 836, 837, 843, 853, 861, 888, 1042, 1043
`\new@glossaryentry` 86, 87, 201, 428, 429, 495, 679, 747, 935, 1002, 1089, 1090, 1158
`\new@ifnextchar` 75, 92, 93, 114–116, 140–144, 146–168, 253–255, 341–345, 422, 432, 433, 445, 463–476, 520, 521, 602–604, 672, 683, 696, 715–727, 772, 773, 855–857, 928, 939, 953, 968–981, 1027, 1028, 1083, 1093, 1094, 1108, 1124–1137, 1183, 1184
`\newacronym` 250, 251, [251](#), 256, 520, 522, 533, 535, 536, 538, 540, 542, 544, 546, 772, 774, 785, 787, 789, 791, 793, 795, 797, 799, 1026, 1027, 1029, 1182, 1184
`\newacronymhook` [255](#), 256, 382, 522, 533, 535, 537, 538, 540, 543, 544, 546, 578, 774, 785, 787, 789, 791, 793, 796, 797, 799, 831, 1029, 1184, 1185
`\newacronymstyle` . 258, [258](#), 259–263, 265, 267, 268, 523–529, 531, 775–781, 783, 1030–1034, 1036–1038, 1186–1190, 1192–1194
`\newcommand` 4–6, 8–33, 35–37, 39–43, 45–48, 50–61, 63–79, 81–93, 96–98, 103–117, 119, 121–126, 128, 130, 137–184, 190, 193–197, 199, 200, 203–209, 211–220, 226–231, 233–235, 237–247, 249–255, 257, 258, 260, 268, 270–272, 274–287, 301, 308,

327, 328, 333, 339–345, 347, 351–355, 372, 382–384, 386– 396, 398–420, 422–435, 439– 446, 449, 450, 452–457, 462– 483, 487, 490–494, 496–503, 505–518, 520–525, 532–549, 553–555, 570, 578, 582, 583, 589, 601–608, 614, 618, 629, 630, 633, 636–646, 648–650, 652–657, 659–664, 666–670, 672–686, 690–697, 700–708, 714–734, 739, 742–746, 748– 777, 784–802, 806–808, 823, 831, 835, 836, 843, 854–861, 868, 872, 883, 886, 888–939, 941, 942, 946–955, 957–959, 961, 968–989, 994, 996–1001, 1004–1030, 1032, 1038–1043, 1045–1067, 1069–1094, 1096, 1097, 1101–1114, 1116, 1122– 1145, 1149, 1152–1155, 1157, 1159–1180, 1182–1187, 1194–1197	399, 401, 402, 500, 635, 643, 648, 651, 652, 752, 889, 898, 903, 905, 907, 951, 1007, 1044, 1052, 1057, 1060, 1062, 1106, 1163
<code>\newcount</code> 17, 84, 391, 427, 641, 677, 895, 933, 1050, 1088	<code>\newignoredglossary</code> 77, 423, 673, 929, 1084
<code>\newcounter</code> 14, 15, 390, 639, 640, 893, 894, 1048, 1049	<code>\newlabel</code> 138, 1123
<code>\NewDocumentCommand</code> 138, 1123	<code>\newlength</code> 137, 290, 291, 302, 314, 321, 331, 462, 609, 615, 622, 626, 631, 714, 862, 863, 869, 875, 876, 879, 880, 885, 968, 1122
<code>\newenvironment</code> 241, 515, 767, 1022, 1178	<code>\newrobustcmd</code> 8, 9, 70–72, 86, 87, 93–96, 114–116, 125, 140– 168, 170, 171, 175, 181, 182, 214, 215, 244, 247, 251–252, 272, 333, 341–344, 355–358, 387, 419–421, 428, 429, 432, 433, 445, 453, 463–482, 502, 503, 520, 521, 555–557, 601– 604, 632, 636, 637, 669–671, 678, 679, 683, 696, 704, 715– 733, 754, 772, 773, 808–810, 854–857, 886, 891, 925, 926, 934, 935, 939–941, 953, 958, 968–983, 985, 988, 1010, 1023, 1025, 1027, 1028, 1039, 1045, 1080, 1081, 1089, 1090, 1093– 1096, 1107, 1108, 1113, 1124– 1138, 1140, 1143, 1144, 1165, 1166, 1179, 1180, 1183, 1184, 1195
<code>\newglossary</code> 19, 20, 40, 41, 75, 76, 77, 198, 392, 393, 404, 405, 422, 423, 493, 642, 643, 655, 672, 673, 745, 896, 897, 910, 928, 929, 1000, 1051, 1052, 1064, 1065, 1083, 1084, 1156	<code>\newterm</code> 41, 405, 655, 910, 1065
<code>\newglossaryentry</code> 9, 41, 83, 86, 86, 112, 256, 387, 405, 426, 428, 443, 522, 532, 534, 535, 537, 539, 541, 543, 545, 578–581, 637, 655, 676–679, 694, 774, 784, 786, 788, 790, 792, 794, 796, 798, 831–834, 891, 910, 932–935, 951, 1029, 1046, 1065, 1087, 1089, 1106, 1185	<code>\newtoks</code> 131, 194, 255, 457, 490, 522, 708, 742, 774, 962, 997, 1028, 1029, 1117, 1152, 1184
<code>\newglossarystyle</code> 246, 284, 287– 300, 302–326, 328–333, 336, 518, 606, 608–634, 770, 859, 861–867, 869–886, 888, 1024, 1180	<code>\newwrite</code> 35, 87, 184, 188, 194, 198, 402, 429, 483, 486, 490, 493, 652, 679, 735, 738, 742, 745, 907, 935, 989, 992, 997, 1000, 1061, 1090, 1145, 1148, 1152, 1156
<code>\newif</code> 6, 7, 22, 29, 33, 36, 111, 209, 385, 386, 394,	<code>\nfss@text</code> 8, 386, 636, 890, 1045
	<code>\noalign</code> 301, 614, 868
	<code>\nobreak</code> 289, 301, 591, 608, 615, 844, 862, 868
	<code>\noexpand</code> 21, 46, 58, 60, 88, 102–104, 107, 128–131, 137, 179, 190–193, 199, 208, 213, 216, 220, 228, 229, 249, 250, 256, 393, 406, 412, 413, 429, 438, 439, 452, 453, 455–457, 462, 481, 487–492, 494, 499,

500, 503, 505, 507, 515, 516,	
520, 522, 532, 534–539, 541–	
543, 545, 549, 578–582, 584,	
643, 657, 663, 680, 689, 690,	
703, 706–708, 713, 714, 732,	
739–745, 751, 752, 755, 757,	
759, 767, 768, 772, 774, 784,	
786–788, 790, 792, 794–796,	
798, 799, 802, 831–835, 837,	
897, 912, 918, 919, 936, 945,	
946, 948, 959–962, 967, 987,	
994–1001, 1006, 1009, 1010,	
1012–1015, 1026, 1029, 1052,	
1067, 1073, 1074, 1090, 1100,	
1101, 1103, 1114–1117, 1122,	
1143, 1149–1156, 1162, 1164,	
1166, 1168, 1170, 1181, 1182, 1185	
noglossaryindex (option)	41
nogroupskip (option)	14
\nohyperpage	247,
519, 770, 771, 1025, 1180, 1181	
nohypertypes (option)	23
\noindent	309–
313, 331–333, 516, 619–621,	
631, 632, 768, 873–875, 885, 886	
\noist	188, 190, 193, 193 , 490,
588, 589, 742, 841, 842, 992,	
993, 996, 997, 1148, 1149, 1152	
nolangwarn (option)	23
nolist (option)	13
nolong (option)	12
nomain (option)	20
nonumberlist (key)	81
nonumberlist (option)	12
\nopostdesc 41 , 50 , 51, 97, 219, 405,	
408, 409, 435, 504, 590, 655,	
659, 686, 756, 843, 910, 914,	
942, 1012, 1065, 1069, 1097, 1167	
nopostdot (option)	13
noredefwarn (option)	24
\normalbaselineskip	301 , 614 , 868
nostyles (option)	13
nosuper (option)	13
notranslate (option)	30
notree (option)	13
nowarn (option)	23
\ns@ACRfull	254 , 521 , 773 , 1028 , 1184
\ns@Acrfull	253 , 521 , 772 , 1028 , 1183
\ns@acrfull	253 , 520 , 772 , 1027 , 1183
\ns@ACRfullpl 255 , 521 , 773 , 1028 , 1184	
\ns@Acrfullpl 254 , 521 , 773 , 1028 , 1184	
\ns@acrfullpl 254 , 521 , 773 , 1028 , 1184	
\ns@ACRlong	166, 475, 726, 980, 1136
\ns@Acrlong	166, 474, 726, 980, 1135
\ns@acrlong	165, 474, 725, 979, 1135
\ns@ACRlongpl 168 , 476 , 727 , 981 , 1137	
\ns@Acrlongpl 168 , 475 , 727 , 981 , 1137	
\ns@acrlongpl 167 , 475 , 727 , 981 , 1136	
\ns@ACRshort	162, 472, 724, 978, 1133
\ns@Acrshort	162, 472, 724, 977, 1133
\ns@acrshort	161, 472, 723, 977, 1133
\ns@ACRshortpl 164 , 473 , 725 , 979 , 1135	
\ns@Acrshortpl 164 , 473 , 725 , 979 , 1134	
\ns@acrshortpl 163 , 473 , 724 , 978 , 1134	
\ns@newglossary	
.	75 , 75 , 422 , 672 , 928 , 1083
\null	130–
137, 190–193, 220, 456–462,	
487–490, 505, 707–714, 739–	
741, 757, 961–967, 994–996,	
1012, 1116–1122, 1149–1152, 1168	
\number	16, 88, 101,
107, 113, 114, 207, 211, 391,	
429, 437, 444, 499, 500, 516,	
549, 641, 680, 688, 695, 751,	
752, 768, 802, 895, 936, 944,	
948, 952, 1006, 1007, 1049,	
1090, 1099, 1103, 1107, 1162, 1163	
numberedsection (option)	10
\numberline	57 , 412 , 662 , 918 , 1072
numberline (option)	10
numbers (option)	40
\numexpr 113 , 114 , 444 , 695 , 952 , 1107	
O	
\O	27, 397, 646, 901, 1055
\o	27, 397, 647, 901, 1055
\OE	27, 397, 646, 901, 1055
\oe	27, 397, 646, 901, 1055
\oldacronym	250 , 520 , 772 , 1026 , 1182
\openout	35 , 87 , 184 , 188 , 194 ,
402, 429, 483, 486, 490, 585,	
588, 652, 679, 735, 738, 742,	
839, 841, 907, 935, 989, 992,	
997, 1061, 1090, 1145, 1148, 1153	
options:	
acronym	20
acronymlists	22
acronyms	20
automake	37

automakegloss	38	smaller	32
automakeglosslite	38	sort	15
compatible-2.07	38	style	11
compatible-3.07	38	subentrycounter	15
counter	22	symbols	40
counterwithin	14	toc	10
debug	7	translate	30
description	32	ucmark	14
disablemakegloss	34	writeglslabelnames	36
dua	32	writeglslabels	36
entrycounter	14	xindy	34
esclocations	12	xindygloss	34
footnote	32	xindynoglsnumbers	34
hyperfirst	31	\OR	545, 798
index	41	\or	7, 9–11, 30, 37, 40, 42, 237, 329, 386, 388, 400, 404, 513, 594, 630, 635–638, 649, 654, 764, 765, 847, 884, 890– 892, 904, 908–911, 1020, 1044– 1047, 1058, 1062, 1064, 1065, 1176
indexonlyfirst	31	order (option)	33
kernelglossredefs	39	\org@glossaryentrynumbers	219, 239, 504, 514, 756, 766, 1012, 1021, 1167, 1177
languages	31	\org@glossarytitle	218, 504, 755, 756, 1011, 1012, 1167
locales	31	\org@glspostdescription	50, 409, 659, 914, 1069
makeindex	33	\org@ifKV@glslink@hyper	127, 128, 454, 455, 705, 706, 959, 1114
mfirstuc	41	\outputpenalty	301, 614, 868
noglossaryindex	41		
nogroupskip	14		
nohypertypes	23		
nolangwarn	23		
nolist	13		
nolong	12		
nomain	20		
nonumberlist	12		
nopostdot	13		
noredefwarn	24		
nostyles	13		
nosuper	13		
notranslate	30		
notree	13		
nowarn	23		
numberedsection	10		
numberline	10		
numbers	40		
order	33		
restoremakegloss	35		
sanitize	29		
sanitizesort	28		
savenuumberlist	12		
savewrites	38		
section	10		
seeautonumberlist	12		
seenoindex	9		
shortcuts	32		
smallcaps	32		
		P	
		\p@	287, 308, 327, 328, 608, 618, 629, 630, 861, 872, 883
		\p@glshyp@opt	124, 125, 453, 704, 958, 1112, 1113
		\PackageError	9, 18, 19, 27, 36, 47, 59, 65, 69, 70, 75, 80, 83, 84, 91–97, 99–101, 112, 123, 170, 193, 194, 198, 201, 203, 226, 229, 230, 232, 236, 238, 245, 246, 251, 257, 258, 358, 384, 385, 387, 392, 403, 407, 413, 417, 419, 422, 424, 426, 427, 431–437, 443, 453, 476, 490, 493, 495–497, 508, 509, 512, 513, 517, 518, 523, 524, 535, 536, 540, 543, 557, 590, 637, 642, 653, 657, 663, 667,

669, 672, 675–677, 682–688,	<code>\PassOptionsToPackage</code>
694, 704, 728, 742, 745, 747– 338, 345, 548, 600, 801, 853
749, 760, 761, 764, 765, 769,	<code>\pdfstringdefPreHook</code> 208, 1007, 1162
770, 775, 776, 788, 789, 793,	<code>\penalty</code> 301, 614, 868
796, 810, 843, 891, 896, 901,	<code>\PGLS</code> 344, 604, 857
907, 912, 919, 923, 925, 928,	<code>\Pgls</code> 342, 342, 603, 856
930, 932, 933, 938–944, 951,	<code>\pgls</code> 341, 342, 602, 855
957, 982, 996, 997, 1000, 1002,	<code>\PGLSpl</code> 344, 604, 857
1004, 1015–1017, 1020, 1021,	<code>\Pglspl</code> 343, 343, 603, 856
1024, 1025, 1027, 1030, 1046,	<code>\pglspl</code> 342, 343, 602, 855
1051, 1056, 1062, 1067, 1073,	<code>\phantomsection</code> 56, 411, 661, 917, 1071
1078–1080, 1083, 1085, 1087,	plural (key) 79
1088, 1093–1096, 1098, 1099,	pluralaccess (key) 347
1106, 1112, 1138, 1152, 1153,	<code>\printacronyms</code> 20, 393, 642, 897, 1052
1156, 1158, 1159, 1171, 1172,	<code>\PrintChanges</code> . 6, 385, 635, 889, 1044
1175, 1176, 1180, 1182, 1185, 1186	<code>\printglossaries</code> 199, 217, 494, 503,
<code>\PackageInfo</code> 7, 34, 35, 194, 206, 386,	746, 755, 1001, 1011, 1157, 1166
402, 490, 498, 635, 636, 651,	<code>\printglossary</code>
652, 742, 750, 890, 906, 997, 20, 24, 40, 41, 199, 216,
1005, 1044, 1045, 1061, 1153, 1161	217, 238, 393, 395, 404, 405,
<code>\PackageWarning</code> 7, 23,	494, 503, 513, 643, 645, 655,
42, 45, 386, 394, 635, 636, 644,	746, 755, 765, 897, 899, 910,
889, 890, 898, 1044, 1045, 1053	1001, 1010, 1011, 1021, 1052,
<code>\PackageWarningNoLine</code> 7, 23, 337,	1054, 1064, 1065, 1157, 1166, 1176
338, 386, 394, 395, 583, 600,	<code>\printindex</code> . . . 41, 405, 655, 910, 1065
635, 636, 644, 836, 853, 889,	<code>\printnoidxglossaries</code>
890, 898, 899, 1044, 1045, 1053 201, 217, 495, 503,
<code>\pagegoal</code> 301, 614, 868	747, 755, 1002, 1011, 1157, 1167
<code>\pagelistname</code>	<code>\printnoidxglossary</code>
. . 47, 50, 294–296, 298, 299,	200, 201, 203, 217, 217, 229,
305, 307, 317–320, 324–327,	230, 238, 494, 495, 497, 503,
407, 408, 611–613, 617, 618,	508, 509, 513, 746, 747, 749,
624–626, 628, 629, 657, 658,	755, 760, 761, 765, 1001, 1002,
864, 865, 867, 870–872, 877–	1004, 1011, 1015, 1016, 1021,
879, 881–883, 912, 914, 1067, 1069	1157, 1159, 1167, 1171, 1172, 1176
<code>\par</code> 51, 287, 289, 308,	<code>\printnumbers</code> . 41, 405, 655, 910, 1065
310–313, 327, 328, 330–336,	<code>\printsymbols</code> . 40, 404, 655, 910, 1064
409, 516, 517, 591, 594–596,	<code>\ProcessOptions</code>
608, 618–622, 629–634, 659, 338, 345, 548, 600, 801, 853
768, 769, 844, 848, 849, 861,	<code>\ProcessOptionsX</code>
862, 872–875, 883–888, 914, 1069 42, 405, 655, 911, 1066
parent (key) 81	<code>\prop</code> 221–223
<code>\parindent</code> 308–313,	<code>\protect</code> . . 57, 122, 171, 258–260,
328, 330–332, 334–336, 594–	266, 282, 363, 364, 373, 374,
596, 618–622, 630–634, 847–	379, 412, 452, 524, 525, 530,
849, 872–875, 883–885, 887, 888	534, 537, 539, 564, 570, 571,
<code>\parskip</code>	575, 662, 702, 703, 776, 777,
308–311, 328, 330, 331, 618–	782, 786, 790, 792, 817, 823,
620, 630–632, 872–874, 883–885	824, 828, 918, 957, 982, 1031,

1032, 1036, 1037, 1042, 1072, 1111, 1112, 1138, 1186–1188, 1192	839, 858, 928, 929, 990, 991, 1000, 1002, 1004, 1005, 1010, 1042, 1082, 1083, 1123, 1145, 1146, 1156, 1157, 1160, 1161, 1166
<code>\protected@csedef</code>	<code>\protected@xdef</code>
. 94, 433, 684, 940, 1095 16–18, 21, 26, 85, 100,
<code>\protected@csxappto</code> 1042	103, 105, 106, 211, 219, 350,
<code>\protected@csxdef</code>	351, 391–393, 396, 427, 440,
. . . 94, 433, 684, 940, 1042, 1094	441, 500, 551–553, 640, 641,
<code>\protected@eappto</code>	643, 646, 678, 688, 690–692,
. . 35, 78, 104, 208, 652, 673, 674, 691, 751, 907, 929, 930, 947, 1006, 1061, 1084, 1101, 1162	752, 756, 805, 806, 895–897, 900, 934, 944, 946–948, 1008, 1012, 1049, 1050, 1052, 1055, 1088, 1098–1102, 1163, 1167, 1168
<code>\protected@edef</code> 8, 11, 18, 21, 35, 60, 61, 64, 66, 67, 70, 74, 76– 78, 94–100, 104, 105, 122, 127– 129, 171, 179, 181, 197, 199, 200, 206, 209, 212, 213, 216, 226, 227, 237, 240, 246, 248, 256, 272, 280, 354, 355, 382, 387, 388, 413, 414, 416, 418, 437, 439, 440, 446–448, 452, 455, 456, 481, 498–500, 513, 515, 516, 518, 522, 541, 545, 549, 554, 555, 585, 586, 601, 605, 636, 638, 641, 643, 652, 663, 665, 667, 668, 670, 672– 674, 684–688, 690, 691, 697– 699, 703, 705–707, 732, 733, 746, 750–752, 755, 757, 758, 765–768, 770, 774, 794, 798, 802, 807, 808, 831, 838, 839, 854, 858, 891, 892, 895–897, 907, 919, 920, 922, 924, 925, 928–930, 940–944, 946, 947, 957, 959, 960, 983, 987, 988, 999, 1001, 1002, 1005, 1007, 1008, 1010, 1013, 1014, 1020, 1021, 1024, 1026, 1029, 1039, 1042, 1045, 1047, 1050, 1052, 1061, 1074, 1075, 1077–1080, 1082–1084, 1094–1098, 1101, 1102, 1111, 1112, 1114, 1115, 1138, 1142–1144, 1155–1157, 1161, 1163, 1164, 1166, 1169, 1176, 1177, 1180, 1181, 1185, 1195	<code>\providecommand</code> 5, 20, 34, 47, 48, 55, 74, 114, 140, 186, 198, 201, 220, 287, 308, 327, 341, 346, 393, 401, 407, 410, 422, 445, 463, 484, 493, 495, 497, 505, 515, 516, 549, 602, 607, 618, 629, 642, 651, 657, 661, 672, 695, 715, 736, 745, 747, 749, 757, 767, 768, 802, 855, 861, 872, 883, 889, 897, 906, 912, 913, 916, 928, 952, 968, 991, 1000, 1002, 1012, 1013, 1043, 1052, 1060, 1067, 1071, 1082, 1107, 1124, 1146, 1156, 1157, 1168
<code>\protected@write</code>	<code>\provideglossaryentry</code>
. . 74, 76, 138, 184–186, 198, 200, 204, 206, 216, 282, 422, 483, 484, 493, 494, 497, 498, 503, 586, 605, 672, 673, 735, 736, 745, 746, 749, 750, 755, 86, 429, 679, 935, 1089
	<code>\ProvidesFile</code> . 48, 407, 657, 913, 1067
	<code>\ProvidesGlossariesLang</code>
 48, 407, 657, 913, 1067
	<code>\ProvidesPackage</code>
	. . 4, 280, 284, 286, 291, 297, 302, 308, 313, 321, 327, 337, 338, 345, 384, 385, 548, 583, 584, 589, 600, 605–607, 609, 613, 615, 618, 622, 626, 629, 635, 801, 836, 837, 843, 853, 858, 859, 861, 863, 866, 869, 872, 875, 879, 883, 888, 1042, 1043
	Q
	<code>\quark</code> 222
	R
	<code>\r</code> 26, 396, 646, 900, 1055
	<code>\raggedright</code>

	300, 302–307, 321–327, 614–618, 626–629, 867–872, 880–883		1161, 1163–1165, 1167, 1168, 1173, 1175, 1176, 1179, 1180, 1195
<code>\raisebox</code>	... 138, 462, 714, 968, 1122	<code>\renewacronymstyle</code> 258, 373–376, 378–381, 523, 570–573, 575–577, 775, 823–826, 828, 830, 1030, 1186
<code>\ref</code> 240, 514, 766, 1022, 1177	<code>\renewcommand</code> 5–7, 9–15, 18–20, 22–24, 28, 30, 31, 34, 37, 38, 40, 42, 45, 49–53, 66, 78, 81, 82, 97, 108, 112, 113, 179, 181, 183, 184, 190, 191, 197–202, 219, 221, 225, 237, 251, 256–268, 284, 285, 288–296, 298, 299, 301–318, 321–326, 328–336, 346, 350, 359, 361, 363, 364, 371–382, 385–390, 392–395, 398–405, 407–409, 417, 423, 425, 435, 443, 444, 481–483, 487, 488, 493–496, 504, 505, 513, 522–533, 535, 536, 538–540, 542–544, 546, 549, 551, 557, 560, 562–564, 569–581, 584, 585, 589–599, 606–624, 626–628, 630–640, 642–645, 647–651, 653, 654, 656, 658, 659, 667, 674–676, 686, 694, 695, 732–734, 739, 740, 745–748, 756, 757, 764, 765, 774–789, 791, 793, 795–797, 799, 802, 804, 805, 810, 813, 815, 817, 822–834, 837, 838, 842–852, 859–878, 880–894, 896–899, 902–904, 906, 908, 910, 911, 913–915, 923, 930–932, 942, 949, 951, 952, 987–989, 994, 995, 1000–1003, 1012, 1013, 1020, 1027, 1029–1038, 1043–1049, 1051–1054, 1056–1059, 1061–1066, 1068–1070, 1078, 1084–1086, 1097, 1103, 1106, 1107, 1143–1145, 1149, 1150, 1155–1159, 1167, 1168, 1176, 1182, 1184–1194
<code>\refstepcounter</code> 106–108, 239, 240, 514, 766, 948, 949, 1021, 1103, 1177	<code>\renewenvironment</code> 241, 284, 288, 291–297, 299, 300, 302–328, 330, 331, 334, 515, 606, 608–612, 614–633, 767, 859, 861, 863–885, 887, 1022, 1178
<code>\regex</code> 244, 1023, 1179	<code>\renewglossarystyle</code> 246, 518, 770, 1024, 1180
<code>\relax</code> 7, 10, 13, 17, 18, 20, 30, 31, 35, 37, 40, 41, 44, 45, 61, 74, 79, 81, 85, 88, 100, 103, 108, 112, 113, 124, 125, 128–130, 132–137, 172, 187, 188, 190, 192, 193, 197–202, 207, 212, 214, 215, 218, 219, 233, 236, 237, 244, 246, 274, 283, 287, 301, 308, 312, 313, 327, 329, 330, 332–336, 349, 350, 364–366, 386, 388, 389, 391–393, 399, 400, 402, 404–406, 414, 422, 424, 425, 428, 429, 437, 439, 441, 444, 453–462, 477, 485–490, 493–496, 499–502, 504, 510, 512, 513, 517, 518, 546, 551, 564, 565, 578–581, 585, 587–589, 594–596, 605, 608, 614, 618, 621, 622, 629–635, 638, 639, 641, 642, 649, 652–656, 664, 672, 674, 675, 678, 680, 688, 690, 692, 694, 695, 704, 706–713, 728, 729, 737–739, 741, 742, 744–748, 751–754, 756, 762, 764, 769, 770, 799, 804, 805, 817, 818, 831–833, 835, 838, 840–842, 847–849, 858, 861, 868, 872, 875, 883–889, 892, 893, 895–897, 903, 904, 907, 909–912, 920, 928, 930, 931, 934, 936, 944, 946, 949, 951, 952, 958, 960–967, 983, 991–994, 996, 997, 999, 1000, 1002, 1003, 1006, 1008–1012, 1017–1020, 1024, 1039, 1042, 1044, 1046, 1048, 1050, 1051, 1058, 1059, 1061, 1062, 1064–1066, 1075, 1082, 1085, 1086, 1089, 1090, 1098–1100, 1104, 1106, 1107, 1112–1122, 1138, 1139, 1147–1152, 1155–1159,		

<code>\RequireGlossariesLang</code>	<code>\seq</code>
..... 48 , 49 , 337 ,	221–223 , 225 , 281 , 283
338 , 407 , 408 , 583 , 600 , 657 ,	<code>\setabbreviationstyle</code>
658 , 836 , 853 , 913 , 914 , 1067 , 1068 273 , 797 , 1039 , 1195
<code>\RequirePackage</code>	<code>\SetAcronymLists</code>
4–6 , 22 , 394 , 643 , 898 , 1052
12 , 13 , 29 , 30 , 43 , 48 , 50 , 273 ,	<code>\SetAcronymStyle</code>
291 , 297 , 302 , 308 , 313 , 321 ,	273 ,
337 , 338 , 346 , 385 , 389 , 399 ,	400 , 401 , 544 , 545 , 650 , 797 , 798
405 , 407 , 546 , 548 , 549 , 583 ,	<code>\setacronymstyle</code>
600 , 609 , 613 , 615 , 618 , 622 ,	251 , 257 , 273 , 523 , 775 , 797 ,
626 , 635 , 638 , 639 , 648 , 649 ,	1027 , 1030 , 1039 , 1182 , 1185 , 1195
655 , 657 , 799 , 801 , 802 , 836 ,	<code>\setbool</code>
853 , 863 , 866 , 869 , 872 , 875 ,	28 , 398 , 648 , 902 , 1057
879 , 888 , 889 , 893 , 903 , 904 ,	<code>\setbox</code>
911 , 913 , 1039 , 1043 , 1044 ,	301 , 614 , 615 , 868
1047 , 1048 , 1058 , 1066 , 1068 , 1195	<code>\setcounter</code>
<code>\restorecounters@</code>	107 , 239 ,
..... 128 , 455 , 706 , 960 , 1115	514 , 766 , 948 , 1021 , 1103 , 1177
<code>restoremakeloss</code> (option)	<code>\SetCustomDisplayStyle</code>
35 273 , 545 , 546 , 798 , 799
<code>\romannumeral</code>	<code>\SetCustomStyle</code>
207 , 211 , 333 , 335 , 499 , 500 ,	273 , 546 , 799
596 , 633 , 634 , 751 , 752 , 849 ,	<code>\SetDefaultAcronymDisplayStyle</code>
886 , 887 , 1006 , 1007 , 1162 , 1163	270 , 532 , 533 , 784 , 785 , 1039 , 1195
S	
<code>\s@glshyp@opt</code>	<code>\SetDefaultAcronymStyle</code>
... 124 , 124 , 453 , 704 , 958 , 1112 270 , 533 , 544 , 785 , 797
<code>\s@GlsSetXdyFirstLetterAfterDigits</code>	<code>\SetDescriptionAcronymDisplayStyle</code>
... 183 , 482 , 734 , 988 , 989 , 1144 271 , 537 , 538 , 789 , 791
<code>\s@GlsSetXdyNumberGroupOrder</code> ..	<code>\SetDescriptionAcronymStyle</code> ...
..... 183 , 271 , 538 , 544 , 791 , 798
184 , 482 , 483 , 734 , 989 , 1144 , 1145	<code>\SetDescriptionDUAAcronymDisplayStyle</code>
<code>\s@ifglossaryexists</code> 271 , 535–537 , 788 , 789
.. 67 , 68 , 217 , 418 , 503 , 668 ,	<code>\SetDescriptionDUAAcronymStyle</code>
669 , 755 , 924 , 1011 , 1079 , 1166 271 , 536 , 544 , 789 , 798
<code>\s@newglossary</code>	<code>\SetDescriptionFootnoteAcronymDisplayStyle</code>
.... 75 , 75 , 422 , 672 , 928 , 1083 271 , 533 , 535 , 786–788
<code>sanitize</code> (option)	<code>\SetDescriptionFootnoteAcronymStyle</code>
29 271 , 535 , 544 , 787 , 798
<code>sanitizesort</code> (option)	<code>\SetDUADisplayStyle</code>
28 272 , 543 , 544 , 796 , 797
<code>\savecounters@</code> 128 , 455 , 706 , 960 , 1114	<code>\SetDUASStyle</code> ..
<code>savenumberlist</code> (option)	272 , 544 , 545 , 797 , 798
12	<code>\setentrycounter</code> 59 , 186 , 235 , 245 ,
<code>savewrites</code> (option)	413 , 484 , 512 , 517 , 584 , 663 ,
38	736 , 764 , 769 , 837 , 919 , 991 ,
<code>sc-short-long</code> (acrstyle)	1019 , 1024 , 1074 , 1146 , 1175 , 1180
261	<code>\SetFootnoteAcronymDisplayStyle</code>
<code>sc-short-long-desc</code> (acrstyle) 263 , 375 271 , 539 , 540 , 791 , 793
<code>section</code> (option)	<code>\SetFootnoteAcronymStyle</code>
10 272 , 540 , 545 , 793 , 798
<code>see</code> (key)	<code>\SetGenericNewAcronym</code>
80 255 , 257 , 522 , 523 ,
<code>seeautonumberlist</code> (option)	774 , 775 , 1029 , 1030 , 1184 , 1186
12	<code>\setglossarypreamble</code>
<code>\seename</code> .. 48 , 214 , 215 , 407 , 502 , 54 , 410 , 660 , 915 , 1070
657 , 754 , 913 , 1010 , 1067 , 1165	
<code>seencindex</code> (option)	
9	

<code>\setglossarysection</code>	<code>\show</code>
..... 55 , 411 , 661 , 916 , 1071 274–279 , 383 , 384 ,
<code>\setglossarystyle</code>	546–548 , 582 , 583 , 799–801 ,
..... 218 ,	835 , 836 , 1040 , 1041 , 1195–1197
237 , 245 , 274 , 288–290 , 292–	<code>\showacronymlists</code>
300 , 303–313 , 315–320 , 322– 278 , 548 , 801 , 1041 , 1197
324 , 326 , 329 , 331 , 332 , 336 ,	<code>\showglocounter</code>
504 , 517 , 518 , 546 , 608–614 , 275 , 546 , 800 , 1040 , 1196
616–632 , 634 , 756 , 769 , 770 ,	<code>\showglodesc</code> 276 , 547 , 800 , 1040 , 1196
799 , 861–886 , 888 , 1011 , 1020 ,	<code>\showglodescaccess</code> 383 , 583 , 836
1024 , 1040 , 1167 , 1175 , 1180 , 1195	<code>\showglodescplural</code>
<code>\setglossentrycompatibility</code> 277 , 547 , 800 , 1041 , 1196
..... 243 , 512 , 516 , 518 , 764 , 768 , 770	<code>\showglodescpluralaccess</code>
<code>\setkeys</code> 383 , 583 , 836
..... 29 , 34 , 45 ,	<code>\showglofield</code> 278 , 548 , 801 , 1041 , 1197
55 , 99 , 127 , 181 , 218 , 256 , 398 ,	<code>\showglofirst</code> 274 , 546 , 800 , 1040 , 1195
401 , 406 , 411 , 436 , 454 , 481 ,	<code>\showglofirstaccess</code> ... 383 , 582 , 835
482 , 504 , 522 , 533 , 535 , 536 ,	<code>\showglofirstpl</code>
538 , 540 , 542 , 544 , 546 , 648 , 275 , 546 , 800 , 1040 , 1196
651 , 656 , 661 , 687 , 705 , 733 ,	<code>\showglofirstpluralaccess</code>
756 , 774 , 785 , 787 , 789 , 791 , 383 , 582 , 836
793 , 795 , 797 , 799 , 903 , 906 ,	<code>\showgloflag</code> 278 , 547 , 801 , 1041 , 1197
912 , 916 , 943 , 959 , 988 , 1011 ,	<code>\showgloindex</code> 278 , 547 , 801 , 1041 , 1197
1029 , 1057 , 1060 , 1066 , 1071 ,	<code>\showglolevel</code> 274 , 546 , 799 , 1040 , 1195
1098 , 1114 , 1143 , 1144 , 1167 , 1184	<code>\showgloloclist</code>
<code>\setlength</code> 278 , 548 , 801 , 1041 , 1197
..... 290 , 291 , 302 , 308–311 , 314 ,	<code>\showglolong</code> 277 , 547 , 801 , 1041 , 1196
321 , 328 , 330 , 331 , 335 , 596 ,	<code>\showglolongaccess</code> 384 , 583 , 836
609 , 615 , 618–620 , 622 , 626 ,	<code>\showglolongpluralaccess</code>
630–634 , 849 , 862 , 863 , 869 , 384 , 583 , 836
872–876 , 879 , 880 , 883–885 , 887	<code>\showglongname</code> 276 , 547 , 800 , 1040 , 1196
<code>\SetSmallAcronymDisplayStyle</code> ..	<code>\showglongnameaccess</code> 383 , 582 , 835
..... 272 , 541–543 , 794–796	<code>\showgloparent</code>
<code>\SetSmallAcronymStyle</code> 274 , 546 , 799 , 1040 , 1195
..... 272 , 542 , 545 , 795 , 798	<code>\showgloplural</code>
<code>\setStyleFile</code> 274 , 546 , 799 , 1040 , 1195
..... 51 , 409 , 659 , 914 , 915 , 1069	<code>\showglopluralaccess</code> .. 383 , 582 , 835
<code>\settodepth</code>	<code>\showgloshort</code> 277 , 547 , 800 , 1041 , 1196
..... 108 , 949 , 1104	<code>\showgloshortaccess</code> ... 384 , 583 , 836
<code>\settoheight</code> . 108 , 462 , 714 , 949 , 1104	<code>\showgloshortpluralaccess</code>
<code>\settowidth</code> 384 , 583 , 836
..... 108 , 260 , 525 ,	<code>\showglosort</code> 277 , 547 , 800 , 1041 , 1196
595 , 596 , 633 , 634 , 777 , 849 ,	<code>\showglossaries</code>
886 , 887 , 949 , 1032 , 1104 , 1187 278 , 548 , 801 , 1041 , 1197
<code>\setupglossaries</code>	<code>\showglossarycounter</code>
..... 45 , 405 , 656 , 911 , 1066 279 , 548 , 801 , 1041 , 1197
<code>\sfcode</code>	<code>\showglossaryentries</code>
..... 13 , 389 , 639 , 893 , 1048 279 , 548 , 801 , 1041 , 1197
<code>short</code> (key)	<code>\showglossaryin</code>
..... 83 279 , 548 , 801 , 1041 , 1197
<code>short-long</code> (acrstyle)	
..... 260 , 373	
<code>short-long-desc</code> (acrstyle) ..	
..... 262 , 375	
<code>shortaccess</code> (key)	
..... 348	
<code>shortcuts</code> (option)	
..... 32	
<code>shortplural</code> (key)	
..... 83	
<code>shortpluralaccess</code> (key)	
..... 348	

<code>\showglossaryout</code>	668, 675–677, 694, 696, 702–704, 732, 735–738, 742–749, 754, 756, 757, 766, 770, 772, 776–784, 790, 794, 817, 823–831, 837–841, 844, 846–852, 859–863, 869, 876, 880, 883–887, 891, 906, 907, 909, 912, 919, 921–924, 928, 931–933, 951, 953, 954, 957, 987, 990–992, 996–1002, 1004, 1006, 1010, 1011, 1013, 1022, 1027, 1031–1039, 1043, 1046, 1060–1062, 1064, 1067, 1073, 1076–1078, 1082, 1086, 1087, 1106, 1108, 1111, 1112, 1142, 1145–1148, 1152–1159, 1162, 1166–1168, 1177, 1178, 1183, 1186–1195
..... 279 , 548 , 801 , 1041 , 1197	
<code>\showglossarytitle</code>	<code>\spacefactor</code> .. 13, 389, 639, 893, 1048
..... 279 , 548 , 801 , 1041 , 1197	<code>\SS</code>
<code>\showglosymbol</code>	27, 397, 647, 901, 1056
..... 277 , 547 , 800 , 1041 , 1196	<code>\ss</code>
<code>\showglosymbolaccess</code> ..	27, 397, 647, 901, 1056
..... 383 , 582 , 836	<code>\str</code>
<code>\showglosymbolplural</code>	245, 1024, 1179
..... 277 , 547 , 800 , 1041 , 1196	<code>\string</code>
<code>\showglosymbolpluralaccess</code>	9, 18, 19, 24, 27, 34–36, 39, 40, 47, 57–64, 66, 74, 76, 81, 83, 89, 92, 93, 104, 105, 112, 114–116, 129, 130, 132–134, 136, 138, 179, 182–190, 192–194, 198–201, 203, 204, 208, 212–214, 218, 220, 221, 229, 230, 238, 251, 273, 282, 384, 385, 387, 392, 395, 401–404, 407, 412–418, 422, 425, 426, 429, 430, 432, 440, 443, 445, 446, 452, 453, 456, 458, 459, 461, 480, 482–487, 489, 490, 493–497, 501, 502, 504, 505, 508, 509, 513, 516–518, 584–589, 605, 637, 642, 645, 651–654, 657, 662, 663, 665–668, 672, 673, 675–677, 680–683, 691, 694–696, 703, 704, 707, 709–711, 713, 732–739, 741, 742, 745–749, 753, 754, 756, 757, 760, 761, 765, 768, 770, 797, 837–842, 858, 891, 896, 899, 901, 906, 907, 909, 912, 918–924, 928, 929, 931–933, 936–939, 947, 951–954, 960, 961, 963, 964, 966, 967, 987–994, 996, 997, 1000–1002, 1004, 1006, 1008, 1009,
..... 383 , 582 , 836	
<code>\showglotext</code> 274 , 546 , 799 , 1040 , 1195	
<code>\showglotextaccess</code>	
..... 383 , 582 , 836	
<code>\showglotype</code> 275 , 546 , 800 , 1040 , 1196	
<code>\showglouserii</code> 275 , 547 , 800 , 1040 , 1196	
<code>\showglouseriii</code>	
..... 276 , 547 , 800 , 1040 , 1196	
<code>\showglouseriv</code>	
..... 276 , 547 , 800 , 1040 , 1196	
<code>\showglouserv</code> 276 , 547 , 800 , 1040 , 1196	
<code>\showglouservi</code>	
..... 276 , 547 , 800 , 1040 , 1196	
<code>sm-short-long</code> (acrstyle)	
..... 261	
<code>sm-short-long-desc</code> (acrstyle) 263 , 376	
<code>smallcaps</code> (option)	
..... 32	
<code>smaller</code> (option)	
..... 32	
<code>\SmallNewAcronymDef</code>	
..... 272 , 382 , 541 , 543 , 581 , 794 , 796 , 834	
<code>sort</code> (key)	
..... 79	
<code>sort</code> (option)	
..... 15	
<code>\space</code>	
..... 9 , 34–36 , 39 , 40 , 47 , 59 , 63 , 64 , 66 , 74 , 81 , 83 , 112 , 114–116 , 121 , 122 , 179 , 185–188 , 193 , 195–199 , 201 , 203 , 208 , 215 , 218 , 221 , 240 , 252 , 258–268 , 272 , 283 , 286 , 288 , 289 , 291 , 303 , 314 , 321 , 322 , 328–330 , 332 , 334 , 335 , 363 , 364 , 373–382 , 387 , 401–404 , 407 , 413 , 416 , 418 , 425 , 426 , 443 , 445 , 446 , 452 , 453 , 480 , 483–486 , 490–497 , 502 , 504 , 505 , 514 , 515 , 518 , 521 , 524–532 , 537 , 541 , 563 , 564 , 570–578 , 584–588 , 590–598 , 606–610 , 615 , 622 , 626 , 630–634 , 637 , 651–654 , 657 , 663 , 666 ,	

1011–1013, 1015, 1016, 1021,	
1027, 1039, 1042, 1046, 1051,	
1054, 1056, 1060–1062, 1064,	
1067, 1072–1078, 1082, 1083,	
1086, 1087, 1091, 1093, 1094,	
1101, 1102, 1106–1108, 1115–	
1119, 1121, 1123, 1142, 1144–	
1149, 1151–1153, 1156–1160,	
1162, 1164, 1165, 1167, 1168,	
1171, 1172, 1176, 1182, 1195	
<code>\strut</code>	288–
291, 293, 295, 303, 304, 306,	
314, 316, 318, 322, 323, 325,	
332, 517, 590–593, 595, 597–	
599, 608–611, 615–617, 622–	
624, 626–628, 632, 769, 844–	
846, 848, 850–852, 861–865,	
869–871, 876–878, 880–882, 886	
style (option)	11
styles:	
<code>altlist</code>	289
<code>altlistgroup</code>	289
<code>altlisthypergroup</code>	289
<code>altlong4col</code>	296
<code>altlong4col-booktabs</code>	299
<code>altlong4colborder</code>	296
<code>altlong4colheader</code>	296
<code>altlong4colheaderborder</code>	297
<code>altlongragged4col</code>	305
<code>altlongragged4col-booktabs</code>	300
<code>altlongragged4colborder</code>	307
<code>altlongragged4colheader</code>	306
<code>altlongragged4colheaderborder</code>	
.	307
<code>altsuper4col</code>	319
<code>altsuper4colborder</code>	320
<code>altsuper4colheader</code>	319
<code>altsuper4colheaderborder</code>	320
<code>altsuperragged4col</code>	325
<code>altsuperragged4colborder</code>	326
<code>altsuperragged4colheader</code>	326
<code>altsuperragged4colheaderborder</code>	
.	326
<code>alttree</code>	333
<code>alttreegroup</code>	336
<code>alttreehypergroup</code>	336
<code>index</code>	328
<code>indexgroup</code>	329
<code>indexhypergroup</code>	329
<code>inline</code>	284
<code>list</code>	287
<code>listdotted</code>	290
<code>listgroup</code>	288
<code>listhypergroup</code>	288
<code>long</code>	291
<code>long-booktabs</code>	297
<code>long3col</code>	292
<code>long3col-booktabs</code>	298
<code>long3colborder</code>	293
<code>long3colheader</code>	294
<code>long3colheaderborder</code>	294
<code>long4col</code>	294
<code>long4col-booktabs</code>	298
<code>long4colborder</code>	295
<code>long4colheader</code>	295
<code>long4colheaderborder</code>	295
<code>longborder</code>	292
<code>longheader</code>	292
<code>longheaderborder</code>	292
<code>longragged</code>	302
<code>longragged-booktabs</code>	299
<code>longragged3col</code>	304
<code>longragged3col-booktabs</code>	300
<code>longragged3colborder</code>	305
<code>longragged3colheader</code>	305
<code>longragged3colheaderborder</code>	305
<code>longraggedborder</code>	303
<code>longraggedheader</code>	303
<code>longraggedheaderborder</code>	304
<code>mcolalmtree</code>	312
<code>mcolalmtreegroup</code>	312
<code>mcolalmtreehypergroup</code>	312
<code>mcolalmtreespannav</code>	313
<code>mcolindex</code>	308
<code>mcolindexgroup</code>	308
<code>mcolindexhypergroup</code>	308
<code>mcolindexspannav</code>	309
<code>mcoltree</code>	309
<code>mcoltreegroup</code>	310
<code>mcoltreehypergroup</code>	310
<code>mcoltreenoname</code>	310
<code>mcoltreenonamegroup</code>	311
<code>mcoltreenonamehypergroup</code>	311
<code>mcoltreenonamespannav</code>	311
<code>mcoltreespannav</code>	310
<code>sublistdotted</code>	290
<code>super</code>	314
<code>super3col</code>	316
<code>super3colborder</code>	316
<code>super3colheader</code>	317

super3colheaderborder	317	superheader (style)	315
super4col	317	superheaderborder (style)	315
super4colborder	318	superragged (style)	321
super4colheader	318	superragged3col (style)	323
super4colheaderborder	319	superragged3colborder (style)	324
superborder	315	superragged3colheader (style)	324
superheader	315	superraggedborder (style)	322
superheaderborder	315	superraggedheader (style)	322
superragged	321	superraggedheaderborder (style)	323
superragged3col	323	superraggedright3colheaderborder	
superragged3colborder	324	(style)	324
superragged3colheader	324	symbol (key)	80
superraggedborder	322	symbolaccess (key)	347
superraggedheader	322	\symbolname	47,
superraggedheaderborder	323	50, 295, 296, 299, 307, 318–	
superraggedright3colheaderborder	324	320, 326, 327, 407, 408, 612,	
tree	330	613, 618, 624–626, 629, 657,	
treegroup	331	658, 865, 867, 871, 872, 878,	
treehypergroup	331	879, 882, 883, 912, 914, 1067, 1069	
treenoname	331	symbolplural (key)	80
treenonamegroup	332	symbolpluralaccess (key)	347
treenonamehypergroup	332	symbols (option)	40
subentrycounter (option)	15	\sys	43, 44
\subglossentry	105, 219, 233,		
243, 285, 288–291, 293, 295,			
303, 304, 306, 314, 316, 318,			
322, 323, 325, 329, 330, 332,			
334, 440, 504, 510, 511, 516,			
517, 607–611, 615–617, 622–			
624, 626–628, 630–633, 691,			
756, 762, 768, 860–865, 869–			
871, 876–878, 880–882, 884,			
885, 887, 947, 1012, 1018,			
1023, 1102, 1167, 1173, 1174, 1179			
\subitem	308, 328, 329, 594,		
618, 629, 630, 847, 872, 883, 884			
sublistdotted (style)	290		
\subsubitem	308, 328, 329,		
594, 618, 630, 847, 872, 883, 884			
super (style)	314		
super3col (style)	316		
super3colborder (style)	316		
super3colheader (style)	317		
super3colheaderborder (style)	317		
super4col (style)	317		
super4colborder (style)	318		
super4colheader (style)	318		
super4colheaderborder (style)	319		
superborder (style)	315		
		T	
		\t	26, 396, 646, 900, 1055
		\tablehead	314–327, 622–629, 876–883
		\tabletail	314–327, 622–629, 876–883
		\tabularnewline	291–296,
		298, 299, 303–307, 314–327,	
		592, 593, 597, 598, 609–613,	
		615–618, 622–629, 846, 850,	
		851, 863–867, 869–872, 876–883	
		\texorpdfstring	170, 208,
		478, 730, 982, 1007, 1138, 1162	
		\text	4, 5, 224, 225, 888, 889, 1043
		text (key)	79
		textaccess (key)	347
		\textbar	283, 606, 859, 1043
		\textbf	249,
		327, 516, 520, 593–596, 629,	
		768, 772, 847–849, 883, 1026, 1182	
		\textit	249, 520, 772, 1026, 1182
		\textmd	249, 520, 772, 1026, 1182
		\textrm	249, 520, 771, 1026, 1182
		\textsc	249, 260, 261, 267, 380,
		520, 526, 531, 535, 538, 540,	
		543, 577, 772, 777, 778, 783,	
		788, 791, 793, 796, 830, 1026,	
		1032, 1033, 1037, 1182, 1188, 1193	

`\textsf` 249, 520, 772, 1026, 1182
`\textsl` 249, 520, 772, 1026, 1182
`\textsmaller` 261, 267, 381, 526, 531,
535, 539, 540, 543, 577, 778,
783, 788, 791, 793, 796, 830,
1032, 1033, 1038, 1188, 1189, 1193
`\texttt` 249, 520, 772, 1026, 1182
`\textulc` . . . 251, 520, 772, 1027, 1183
`\textup` 249, 251,
520, 772, 1026, 1027, 1182, 1183
`\TH` 27, 397, 647, 901, 1056
`\th` 27, 397, 647, 901, 1056
`\the` 46, 49, 59,
64, 66, 76, 132–137, 184, 188,
190–192, 204–206, 211, 216,
233, 248, 249, 256–260, 264,
266, 373, 377, 379, 382, 406,
408, 413, 416, 418, 422, 457–
462, 483, 486, 488, 489, 497,
498, 500, 503, 510, 515, 516,
519, 522–525, 529, 530, 532–
543, 545, 549, 570, 571, 574,
575, 578–582, 584, 585, 588,
657, 658, 663, 667, 668, 672,
708–714, 735, 738–741, 749,
750, 752, 755, 762, 767, 768,
771, 774–777, 780, 782, 784–
788, 790, 792, 794–796, 798,
799, 802, 823, 824, 827, 828,
831–835, 837, 839, 841, 912,
913, 919, 922, 924, 928, 962–
967, 989, 992, 994, 996, 1004,
1005, 1007, 1010, 1018, 1026,
1029–1032, 1035, 1036, 1067,
1068, 1073, 1077, 1078, 1083,
1117–1122, 1145, 1148, 1150–
1152, 1160, 1161, 1163, 1166,
1173, 1181, 1185–1187, 1191, 1192
`\the@numberlist`
179, 180, 481, 732, 733, 987, 1143
`\theglossary` . . 6, 385, 635, 889, 1044
`\theglossary (env.)` 241
`\theglossaryentry` 14, 240, 390, 514,
639, 766, 894, 1022, 1048, 1177
`\theglossarysubentry`
. 15, 240, 390, 515,
640, 766, 894, 1022, 1049, 1178
`\theglentrycounter` 128, 129, 209,
211, 212, 455, 499, 500, 585,
706, 751, 752, 838, 959, 960,
1007, 1008, 1114, 1115, 1162, 1163
`\theH` 213, 501, 753, 1009, 1165
`\theHglossaryentry`
. 14, 390, 639, 894, 1048
`\theHglossarysubentry`
. 15, 390, 640, 894, 1049
`\theHglentrycounter`
. . . . 128, 129, 209, 212, 455,
456, 499, 500, 706, 751, 752,
960, 1007, 1008, 1115, 1162, 1163
`\thepage` 138, 1123
`\thesection` . . . 46, 406, 656, 912, 1067
`\this@dialect` 49, 50, 337, 338, 408,
583, 600, 658, 836, 853, 914, 1068
`\tiny` 8, 387, 636, 890, 1045
`\tl` 5, 44, 107, 138,
222–225, 889, 1043, 1103, 1123
`toc (option)` 10
`\toks@` 46, 49, 59, 64, 66, 76,
131–137, 190–192, 216, 248,
249, 406–408, 413, 416, 418,
422, 457–462, 487–489, 503,
515, 516, 519, 549, 584, 657,
658, 663, 667, 668, 672, 708–
714, 739–741, 755, 767, 768,
771, 802, 837, 912, 913, 919,
922, 924, 928, 962–967, 994,
996, 1010, 1025, 1026, 1067,
1068, 1073, 1077, 1078, 1083,
1117–1122, 1150–1152, 1166, 1181
`\toprule` 298, 299, 613, 866, 867
`\TrackIfKnownLanguage` . 48, 913, 1068
`\trans@languages`
. 49, 407, 657, 913, 1068
`\translate` 49, 50,
408, 658, 659, 913, 914, 1068, 1069
`translate (option)` 30
`\translatelet` . 19, 20, 40, 41, 392,
393, 404, 405, 642, 643, 655,
896, 897, 910, 1051, 1052, 1065
`tree (style)` 330
`treegroup (style)` 331
`treehypergroup (style)` 331
`treenoname (style)` 331
`treenonamegroup (style)` 332
`treenonamehypergroup (style)` 332
`\triangleright` 8, 387, 636, 890, 1045
`\ttfamily` 8, 387, 636, 890, 1045
`\TX@trial` . . . 108, 441, 692, 949, 1104

type (key)	80	<code>\warn@noprintglossary</code> .	199–201 ,
<code>\typeout</code>	23 , 394 , 644 , 898 , 1053		216 , 219 , 493–495 , 503 , 504 ,
U			
<code>\u</code>	26 , 396 , 646 , 900 , 1055	<code>\write</code>	36 , 44 , 89 , 114 , 184–
<code>\uccode</code>	233 , 510 , 762 , 1017 , 1173		189 , 201 , 205 , 220 , 402 , 429 ,
ucmark (option)	14		430 , 445 , 483–487 , 491 , 492 ,
<code>\undef</code>	82 , 88 , 216 ,		495 , 498 , 505 , 585–589 , 652 ,
	425 , 429 , 503 , 676 , 680 , 755 ,		680 , 695 , 696 , 735–739 , 743 ,
	932 , 936 , 1010 , 1086 , 1090 , 1166		744 , 747 , 749 , 757 , 839–842 ,
<code>\unexpanded</code>	42 , 911 , 1065		907 , 936 , 952 , 953 , 989–993 ,
<code>\unskip</code>	97 , 290 , 435 ,		997–1000 , 1002 , 1005 , 1012 ,
	591 , 609 , 686 , 844 , 862 , 942 , 1097		1013 , 1062 , 1091 , 1107 , 1145–
<code>\unvbox</code>	301 , 614 , 615 , 868		1149 , 1153–1155 , 1157 , 1160 , 1168
<code>\use</code>	226	<code>writelsgllabelnames</code> (option)	36
<code>\usedictionary</code>	49 , 407 , 658 , 913 , 1068	<code>writelsgllabels</code> (option)	36
<code>\usepackage</code>		<code>\writeist</code>	184 ,
	27 , 229 , 230 , 251 , 384 , 385 ,		193 , 194 , 483 , 486 , 487 , 490 ,
	508 , 509 , 760 , 761 , 901 , 1015 ,		585 , 588 , 589 , 735 , 737 , 739 ,
	1016 , 1027 , 1056 , 1171 , 1172 , 1182		742 , 839 , 841 , 842 , 989 , 992 ,
user1 (key)	82		993 , 997 , 1145 , 1148 , 1149 , 1152
user1access (key)	348	X	
user2 (key)	82	<code>\x</code>	519
user2access (key)	348	<code>\xatlevel@</code>	128 , 455 , 706 , 959 , 1114
user3 (key)	82	<code>\xcapitalisewords</code>	478
user3access (key)	348	<code>\xdef</code>	101 , 437 , 439 ,
user4 (key)	83		504 , 605 , 688 , 858 , 859 , 944 , 1099
user4access (key)	349	<code>\xglsaccsupp</code>	355 , 355 , 554 , 555 , 808
user5 (key)	83	<code>\xglsfieldaccsupp</code>	354 , 554 , 807
user5access (key)	349	<code>\xifinlist</code>	87 , 679 , 935 , 1090
user6 (key)	83	<code>\xifinlistcs</code>	227 ,
user6access (key)	349		228 , 231 , 506 , 509 , 757 , 758 ,
<code>\UTFviii@two@octets</code>			761 , 1013 , 1014 , 1016 , 1169 , 1172
	27 , 397 , 647 , 901 , 1056	<code>xindy</code> (option)	34
<code>\UTFviii@two@octets@combine</code>		<code>xindygloss</code> (option)	34
	27 , 397 , 647 , 901 , 1056	<code>xindynoglsnumbers</code> (option)	34
V			
<code>\v</code>	26 , 397 , 646 , 901 , 1055	<code>\xmakefirstuc</code>	
<code>\value</code>	107 , 948 , 1103		446–448 , 452 , 476 , 477 ,
<code>\vbox</code>	301 , 614 , 615 , 868		601 , 697–699 , 703 , 728 , 729 , 854
<code>\vsize</code>	301 , 615 , 868	<code>\xspace</code>	250 , 520 , 772 , 1026 , 1182
<code>\vskip</code>	287 , 301 , 308 , 327 , 608 ,	Y	
	614 , 618 , 629 , 861 , 868 , 872 , 883	<code>\year</code>	184 ,
<code>\vss</code>	301 , 614 , 615 , 868		188 , 483 , 486 , 585 , 588 , 735 ,
W			
<code>\warn@nomakeglossaries</code>			738 , 839 , 841 , 989 , 992 , 1145 , 1148
	197 , 199–201 , 493–495 ,	Z	
	744–747 , 999–1002 , 1155–1158	<code>\Z</code>	244 , 1023 , 1179
		<code>\z@</code>	301 , 614 , 615 , 868