Documented Code For glossaries v4.54

Nicola L.C. Talbot

Dickimaw Books
http://www.dickimaw-books.com/

2024-04-03

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 44
2.4 Xindy 54
2.5 Loops and conditionals 63
2.6 Defining new glossaries 70
2.7 Defining new entries 75
2.8 Resetting and unsetting entry flags 103
2.9 Keeping Track of How Many Times an Entry Has Been Unset . . 108
2.10 Loading files containing glossary entries 113
2.11 Using glossary entries in the text 114
2.11.1 Links to glossary entries 119
2.11.2 Displaying entry details without adding information to the glossary 166
2.12 Adding an entry to the glossary without generating text 177
2.13 Creating associated files 179
2.14 Writing information to associated files 200
2.15 Glossary Entry Cross-References 210
2.16 Displaying the glossary 212
2.17 Acronyms 241
2.18 Predefined acronym styles 246
2.19 Predefined Glossary Styles 264
2.20 Debugging Commands 265
2.21 Compatibility with version 2.07 and below 270

3 Glossary Styles **271**
3.1 Glossary hyper-navigation definitions (glossary-hypernav.sty) . . 271
3.2 In-line Style (glossary-inline.sty) 275
3.3 List Style (glossary-list.sty) 277
3.4 Glossary Styles using longtable (glossary-long.sty) 281
3.5 Glossary Styles using longtable and booktabs (glossary-longbooktabs.sty) 288
3.6 Glossary Styles using longtable (glossary-longragged.sty) 293
3.7 Glossary Styles using multicol (glossary-mcols.sty) 298
3.8 Glossary Styles using supertabular environment (glossary-super.sty) 304
3.9 Glossary Styles using supertabular environment (glossary-superragged.sty) 311
3.10 Tree Styles (glossary-tree.sty) 318

4 Multi-Lingual Support **327**
4.1 Babel Captions (glossaries-babel.sty) 327
4.2 Polyglossia Captions (glossaries-polyglossia.sty) 328

5 Prefix Support (glossaries-prefix Code) **329**

6 Accessibility Support (glossaries-accsupp Code) **336**
6.1 Defining Replacement Text 338
6.2 Accessing Replacement Text 342
6.3 Displaying the Glossary 362
6.4 Acronyms 364
6.5 Debugging Commands 374

7 Backwards Compatibility **375**
7.1 glossaries-compatible-207 375
7.2 glossaries-compatible-307 375

8 Rollback **376**
8.1 Rollback v4.46 (glossaries-2020-03-19.sty) 376
8.2 Rollback v4.46 (glossaries-accsupp-2020-03-19.sty) 539
8.3 Rollback v4.46 (glossaries-babel-2020-03-19.sty) 574
8.4 Rollback v4.46 (glossaries-compatible-207-2020-03-19.sty) 575
8.5 Rollback v4.46 (glossaries-compatible-307-2020-03-19.sty) 580
8.6 Rollback v4.46 (glossaries-polyglossia-2020-03-19.sty) 591
8.7 Rollback v4.46 (glossaries-prefix-2020-03-19.sty) 591
8.8 Rollback v4.46 (glossary-hypernav-2020-03-19.sty) 596
8.9 Rollback v4.46 (glossary-inline-2020-03-19.sty) 597
8.10 Rollback v4.46 (glossary-list-2020-03-19.sty) 598
8.11 Rollback v4.46 (glossary-list-2020-03-19.sty) 600
8.12 Rollback v4.46 (glossary-longbooktabs-2020-03-19.sty) 603
8.13 Rollback v4.46 (glossary-longragged-2020-03-19.sty) 606
8.14 Rollback v4.46 (glossary-mcols-2020-03-19.sty) 609
8.15 Rollback v4.46 (glossary-super-2020-03-19.sty) 613
8.16 Rollback v4.46 (glossary-superragged-2020-03-19.sty) 617

8.17 Rollback v4.46 (glossary-tree-2020-03-19.sty) 620
8.18 Rollback v4.49 (glossaries-2021-11-01.sty) 626
8.19 Rollback v4.49 (glossaries-accsupp-2021-11-01.sty) 792
8.20 Rollback v4.49 (glossaries-babel-2021-11-01.sty) 827
8.21 Rollback v4.49 (glossaries-compatible-207-2021-11-01.sty) 828
8.22 Rollback v4.49 (glossaries-compatible-307-2021-11-01.sty) 834
8.23 Rollback v4.49 (glossaries-polyglossia-2021-11-01.sty) 844
8.24 Rollback v4.49 (glossaries-prefix-2021-11-01.sty) 844
8.25 Rollback v4.49 (glossary-hypernav-2021-11-01.sty) 849
8.26 Rollback v4.49 (glossary-inline-2021-11-01.sty) 850
8.27 Rollback v4.49 (glossary-list-2021-11-01.sty) 852
8.28 Rollback v4.49 (glossary-long-2021-11-01.sty) 854
8.29 Rollback v4.49 (glossary-longbooktabs-2021-11-01.sty) 857
8.30 Rollback v4.49 (glossary-longragged-2021-11-01.sty) 860
8.31 Rollback v4.49 (glossary-mcols-2021-11-01.sty) 863
8.32 Rollback v4.49 (glossary-super-2021-11-01.sty) 866
8.33 Rollback v4.49 (glossary-superragged-2021-11-01.sty) 870
8.34 Rollback v4.49 (glossary-tree-2021-11-01.sty) 874
8.35 Rollback v4.52 (glossaries-2022-11-03.sty) 879
8.36 Rollback v4.52 (glossary-hypernav-2022-11-03.sty) 1033

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 v4.54: LaTeX2e 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}
\DeclareCurrentRelease{v4.54}{2024-04-03}
```

Declare package:

```
\ProvidesPackage{glossaries}[2024/04/03 v4.54 (NLCT)]
```

Required packages:

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

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

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

\ExplSyntaxOn
```

`\glslowercase` Provided to make it easier for user to use LaTeX3 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]{
  \tl_if_in:NnF \l_text_case_exclude_arg_tl {#1}
   {
     \tl_put_right:Nn \l_text_case_exclude_arg_tl {#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 \glsuppercase 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}

\RequirePackage{datatool-base}
```

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@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
}
```

`\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]{%
  \ifstrempty{#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@choicekey{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@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
}
```

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 ignores 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]{}
\glsesclocationsfalse
```

\@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 use upper case in definition of \glsglossarymark

```
\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.

fine@glossarysubentrycounter

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

Check if counter already defined.

```
  \ifundef\c@glossarysubentry
  {%
    \ifglssubentrycounter
      \ifglsentrycounter
        \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}{\ifglsxindy 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

\glsprestandardsort{⟨*sort cs*⟩}{⟨*type*⟩}{⟨*label*⟩}

Allow user to hook into sort mechanism. The first argument ⟨*sort cs*⟩ 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
```

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

```
\newcommand*\glssortnumberfmt[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@xdef\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 glossary 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\glssortnumberfmt
           {\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][]{%
      \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}{}
```

\@addtoacronynlists

```
\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

\glsIfListOfAcronyms{⟨*label*⟩}{⟨*true part*⟩}{⟨*false part*⟩}

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@descplural@field
```
\glssetnoexpandfield{descplural}
```
\@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}
```
ls@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
    \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
}
```

sanitizesort

```
\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
}
```

Default setting:

```
\glssanitizesorttrue
\glssetnoexpandfield{sortvalue}%
```

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

```
\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}%
}
```

`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}%
}
```

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

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

`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}%
      }%
    }%
  }%
}
```

`restoremakegloss` Cancel the effect of disablemakegloss.

```
\@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
    }%
  }%
}
```

\@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.

```
\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. 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

```
   \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}%
  \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}%
  \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]{%
    \ifglssavewrites
      \renewcommand*{\glswritefiles}{\@glswritefiles}%
    \else
      \let\glswritefiles\@empty
    \fi
  }
```

Set default:

```
  \glssavewritesfalse
  \let\glswritefiles\@empty
```

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

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

```
\ifglsautomake
 \ifdef\directlua{\RequirePackage{shellesc}}{}
\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,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
  \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
}
```

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}
```

\glssettoctitle Sets the TOC title for the given glossary.

```
\newcommand*{\glssettoctitle}[1]{%
 \def\glossarytoctitle{\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 translations 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, multlingual 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*{\glssymbolsgroupname}{%
      \translate{Symbols (glossaries)}}%
    \renewcommand*{\glsnumbersgroupname}{%
      \translate{Numbers (glossaries)}}%
  }{}%
\fi
```

`\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.

```
  \ifglsxindy
    \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

```
\ifglsxindy
  \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 LaTeX 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 `\glossarypremable` 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`.

\glossarypreamble

```
\newcommand*{\glossarypreamble}{%
  \csuse{@glossarypreamble@\currentglossary}%
}
```

\setglossarypreamble

```
\setglossarypreamble[⟨type⟩]{⟨text⟩}
```

Code provided by Michael Pock.

```
\newcommand{\setglossarypreamble}[2][\glsdefaulttype]{%
  \ifglossaryexists*{#1}%
```

```
    {\csgdef{@glossarypreamble@#1}{#2}}%
    {\GlossariesWarning{Glossary `#1' is not defined}}%
  }
```

The glossary postamble is given by `\glossarypostamble`. 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{\glossarypostamble}{For a complete list of terms
%see \cite{blah}\gdef\glossarypreamble{}}
%
```

`\glossarypostamble`

```
\newcommand*{\glossarypostamble}{}
```

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

```
\newcommand*{\glossarysection}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@p@glossarysection{#1}{#2}%
  }%

  \glsglossarymark{\glossarytoctitle}%
}
```

`\glsglossarymark` Sets the header mark for the glossary. Takes the glossary short (TOC) title as the argument. If `\glossarymark` 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{\memUChead{#1}}{\memUChead{#1}}%
      \else
```

```
        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand{\glsglossarymark}[1]{%
      \ifglsucmark
        \@mkboth{\glsuppercase{#1}}{\glsuppercase{#1}}%
      \else
        \@mkboth{#1}{#1}%
      \fi
    }
  }
}
```

\glossarymark Provided for backward compatibility:

```
\providecommand{\glossarymark}[1]{%
  \ifglsucmark
    \@mkboth{\glsuppercase{#1}}{\glsuppercase{#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]{%
  \glsclearpage
  \phantomsection
  \ifdefempty\@@glossarysecstar
  {%
    \csname\@@glossarysec\endcsname{#2}%
  }%
  {%
    \@gls@toc{#1}{\@@glossarysec}%
      \csname\@@glossarysec\endcsname*{#2}%
  }%
```

Do automatic labelling if required

```
  \@@glossaryseclabel
}
```

\gls@doclearpage The \gls@doclearpage 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@doclearpage}{%
  \ifthenelse{\equal{\@@glossarysec}{chapter}}%
  {%
    \ifcsundef{cleardoublepage}%
    {%
      \clearpage
    }%
    {%
      \ifcsdef{if@openright}%
      {%
         \if@openright
           \cleardoublepage
         \else
           \clearpage
         \fi
      }%
      {%
         \cleardoublepage
      }%
    }%
  }%
  {}%
}
```

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

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

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)

```
\ifglsxindy
  \edef\@xdyattributes{\string"default\string"}%
\fi
```

\@xdyattributelist Comma-separated list of attributes.

```
\ifglsxindy
  \edef\@xdyattributelist{}%
\fi
```

\@xdylocref Define list of markup location references.

```
\ifglsxindy
  \def\@xdylocref{}
\fi
```

\@gls@ifinlist

```
\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
}
```

\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.

```
\ifglsxindy
  \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{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
```

`\@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.

`y@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"%
}%
```

`y@Lclass@arabic-page-numbers` Arabic numbers (1, 2, ...).

```
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
  \string"arabic-numbers\string"%
}%
```

`dy@Lclass@alpha-page-numbers` Lower case alphabetical (a, b, ...).

```
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
  \string"alpha\string"%
}%
```

`dy@Lclass@Alpha-page-numbers` Upper case alphabetical (A, B, ...).

```
\expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
  \string"ALPHA\string"%
}%
```

`Lclass@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.

```
\ifglsxindy
 \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.)

```
\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"}%
   }
```

Only has an effect before `\writeist`:

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

`\@xdylocationclassorder` Define location class order

```
\ifglsxindy
  \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

```
\ifglsxindy
  \newcommand*\GlsSetXdyLocationClassOrder[1]{%
    \def\@xdylocationclassorder{#1}}
\else
  \newcommand*\GlsSetXdyLocationClassOrder[1]{%
    \glsnoxindywarning\GlsSetXdyLocationClassOrder}
\fi
```

\@xdysortrules Define sort rules

```
\ifglsxindy
  \def\@xdysortrules{}
\fi
```

\GlsAddSortRule Add a sort rule

```
\ifglsxindy
  \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]{%
    \glsnoxindywarning\GlsAddSortRule}
\fi
```

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

```
\ifglsxindy
  \def\@xdyrequiredstyles{tex}
\fi
```

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

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

\GlsSetXdyStyles Reset the list of required styles

```
\ifglsxindy
  \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.

```
\ifglsxindy
  \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.

```
\ifglsxindy
  \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]{%
```

```
    \glsnoxindywarning\GlsSetXdyCodePage}
\fi
```

`\@xdylettergroups` Store letter group definitions.

```
\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
```

`\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}%
}
```

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

```
\forglsentries[⟨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*{\forglsentries}[3][\glsdefaulttype]{%
  \protected@edef\@@glo@list{\csname glolist@#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@}%
  {%
    \forglsentries[\@@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 `\glsentrytext` 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}
```

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

\ifglsentryexists{⟨*label*⟩}{⟨*true text*⟩}{⟨*false text*⟩}

where ⟨*label*⟩ is the entry's label.

```
\newcommand{\ifglsentryexists}[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{⟨*label*⟩}{⟨*code*⟩}

\glsdoifexists Generate an error if entry specified by ⟨*label*⟩ doesn't exists, 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 exists. 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 exists, 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 exists, 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\forglsentries\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}%
  {%
    \ifcsempty{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
  }%
}
```

\ifglshasdesc

\ifglshasdesc{⟨*label*⟩}{⟨*true part*⟩}{⟨*false part*⟩}

```
\newcommand*{\ifglshasdesc}[3]{%
  \ifcsempty{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

`\ifglsfieldvoid{⟨field⟩}{⟨label⟩}{⟨true part⟩}{⟨false part⟩}`

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

```
\newcommand{\ifglsfieldvoid}[4]{\ifcsvoid{glo@\glsdetoklabel{#2}@#1}{#3}{#4}}
```

\ifglshasfield

\ifglshasfield{⟨*field*⟩}{⟨*label*⟩}{⟨*true part*⟩}{⟨*false part*⟩}

```
\newrobustcmd*{\ifglshasfield}[4]{%
  \glsdoifexists{#2}%
  {%
    \letcs{\@glo@thisvalue}{glo@\glsdetoklabel{#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@forbidtexext 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@forbidtexext}[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\glsdefaulttype
  {%
    \gdef\glsdefaulttype{#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@forbidtexext\csname @glotype@#2@log\endcsname
    \expandafter\@gls@forbidtexext\csname @glotype@#2@in\endcsname
    \expandafter\@gls@forbidtexext\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 LaTeX, `\@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 @glotype@#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 @glotype@#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 ⟨*name*⟩ ⟨*description*⟩.

```
\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 `\glsentryfmt` (or use for `\defglsentryfmt` 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

\gls@set@xr@key{⟨*key name*⟩}{⟨*cs*⟩}{⟨*value*⟩}

Assign a cross-reference key.

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

\gls@xr@key

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

\gls@checkseeallowed

```
\newcommand*{\gls@checkseeallowed}{%
```

```
 \@gls@see@noindex
}
```

`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@userv{#1}%
}
```

user6

```
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#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]{%
  \ifglsentryexists{#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.

```
  \ifglsentryexists{#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
  \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:

```
  \glswritedefhook
  \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

```
\glsaddkey{⟨key⟩}{⟨default value⟩}{⟨no link cs⟩}{⟨no link ucfirst
cs⟩}{⟨link cs⟩}{⟨link ucfirst cs⟩}{⟨link allcaps cs⟩}
```

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][]{%
          \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}%
    }%
```

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}{}%
   }%
 }%
}
```

\glsfieldedef

\glsfieldedef{⟨*label*⟩}{⟨*field*⟩}{⟨*definition*⟩}

```
\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}{}%
   }%
 }%
}
```

\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

\glsunexpandedfieldvalue{⟨*label*⟩}{⟨*field*⟩}

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.

```
\newcommand*{\glsunexpandedfieldvalue}[2]{%
  \ifcsname glo@\glsdetoklabel{#1}@#2\endcsname
   \expandafter\expandonce\csname glo@\glsdetoklabel{#1}@#2\expandafter\endcsname
  \fi
}
```

\ifglsfieldeq

\ifglsfieldeq{⟨*label*⟩}{⟨*field*⟩}{⟨*string*⟩}{⟨*true*⟩}{⟨*false*⟩}

Tests if the value of the given field is equal to the given string.

```
\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}{}%
   }%
 }%
}
```

\ifglsfielddefeq

\ifglsfielddefeq{⟨*label*⟩}{⟨*field*⟩}{⟨*command*⟩}{⟨*true*⟩}
{⟨*false*⟩}

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}{}%
   }%
 }%
}
```

\glswritedefhook

```
\newcommand*{\glswritedefhook}{}
```

\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]{%
  \ifglsentryexists{#1}{}%
  {\longnewglossaryentry{#1}{#2}{#3}}%
}
\@onlypreamble{\longprovideglossaryentry}
```

\gls@defglossaryentry

```
\gls@defglossaryentry{⟨label⟩}{⟨key-val list⟩}
```

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\@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
```

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 nonumberlist setting if we're in the document environment.

```
\@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{}%
```

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\@glolist@{\csname glolist@\@glo@type\endcsname}%

    \expandafter\protected@xdef\csname glolist@\@glo@type\endcsname{%
      \@glolist@{\@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{@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\@gloname\@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\@glodesc\@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@glssee{%
      \noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
      \noexpand\expandafter\noexpand\@glssee\noexpand\@glo@list{\@glo@label}}%
    \@do@glssee
  }%
  \@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{,}%
  \forglsentries[\glo@type]{\glo@label}%
   {%

     \ifdefequal\@glo@thislabel\glo@label
       {}{\protected@eappto\glo@list{\glo@label,}}%
   }%
  \cslet{glolist@\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.)

```
\ifglsxindy
  \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.)

```
\ifglsxindy
  \newcommand*{\@glossarysubentryfield}{%
    \string\\subglossentry}
\else
  \newcommand*{\@glossarysubentryfield}{%
    \string\subglossentry}
\fi
```

\@glo@storeentry

\@glo@storeentry{⟨*label*⟩}

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.

```
  \ifglsxindy
```

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*{\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{}%
}
```

\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 LaTeX counter or even an explicit TeX 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.

ewglossaryentry@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{glo@\glsdetoklabel{##1}@currcount}%
 {0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*{\glsentryprevcount}[1]{%
 \ifcsundef{glo@\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{glo@\glsdetoklabel{##1}@currcount}{0}\fi
}%
\renewcommand*{\@glslocalreset}[1]{%
  \@@glslocalreset{##1}%
  \ifglsresetcurrcount\csdef{glo@\glsdetoklabel{##1}@currcount}{0}\fi
}%
```

Alter behaviour of \cgls. (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\@cgls@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cglsformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@gls@{##1}{##2}[##3]%
  \fi
}%
```

Similarly for the analogous commands. No case change plural:

```
\def\@cglspl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cglsplformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@glspl@{##1}{##2}[##3]%
  \fi
}%
```

First letter uppercase singular:

```
\def\@cGls@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cGlsformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@Gls@{##1}{##2}[##3]%
  \fi
}%
```

First letter uppercase plural:

```
\def\@cGlspl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cGlsplformat{##2}{##3}%
    \glsunset{##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{glo@\glsdetoklabel{##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{glo@\glsdetoklabel{#1}@currcount}{%
   \number\numexpr\glsentrycurrcount{#1}+1}%
}
```

`ls@local@increment@currcount`

```
\newcommand*{\@gls@local@increment@currcount}[1]{%
  \csedef{glo@\glsdetoklabel{#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]{}
```

`\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}
```

`\@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]{%
  \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]{%
  \ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}#2%
}
```

`\cglspl` 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*{\cglspl}{\@gls@hyp@opt\@cglspl}
```

`\@cglspl` 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%
}
```

`\cGlspl` 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*{\cGlspl}{\@gls@hyp@opt\@cGlspl}
```

Add sentence case mapping.

```
\glsmfuaddmap{\cglspl}{\cGlspl}
```

`\@cglspl` 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 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}
```

[1] and any other valid LaTeX code that can be used in the preamble.

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
  \glscapscase
  {%
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:

```
        \glscapscase
        {%
```

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:
```
            \glsuppercase
              {\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert}%
          }%
        }%
        {%
```
Subsequent singular form
```
          \glscapscase
          {%
```
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:
```
            \glsuppercase
              {\acronymfont{\glsentryshort{\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

```
          \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

\genacrfullformat{⟨*label*⟩}{⟨*insert*⟩}

The full format used by \glsgenacfmt (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\glssentencecase\expandafter{\gls@text}%
}
```

\genplacrfullformat

\genplacrfullformat{⟨*label*⟩}{⟨*insert*⟩}

The full format used by \glsgenacfmt (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\glssentencecase\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 LaTeX 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}}
```

```
ls@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}{\glssentencecase{#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 \glstype and the label is stored in \glslabel.

```
\newcommand*{\@gls@link@checkfirsthyper}{%
  \ifglsused{\glslabel}%
  {%
  }%
```

```
    {%
      \gls@checkisacronymlist\glstype
      \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{\glstype}{\@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\theHglsentrycounter{\@gls@Hcounter}%
        \restorecounters@
      \fi
    }%
  }%
  {%
```

Not using equation counter so no special measures:

```
    \edef\theglsentrycounter{\expandafter\noexpand
      \csname the\@gls@counter\endcsname}%
  }%
```

Check if hyperref version of this counter

```
  \ifx\@gls@Hcounter\@empty
    \ifcsundef{theH\@gls@counter}%
    {%
```

```
      \def\theHglsentrycounter{\theglsentrycounter}%
    }%
    {%
      \protected@edef\theHglsentrycounter{\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@mkidxrangechar#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@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}
```

\@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 \glsromanpage (thanks to David Carlise for the suggestion.) Skip if esclocations=false.

```
  \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@dosubst{%
        \noexpand\DTLsubstituteall\noexpand\gls@xdystring
        {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
      }%
      \gls@dosubst
    }%
  \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]{%
  \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\null
    }%
    \@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\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
}
```

`\@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\@glstarget\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\glscapscase\@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
}
```

\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\glscapscase\@secondofthree
    \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
}
```

\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\glscapscase\@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 `\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 `\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\glscapscase\@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\glscapscase\@secondofthree
    \let\glscustomtext\@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 \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 like \glspl except that all the link text is converted to uppercase.

\GLSpl

```
\newrobustcmd*{\GLSpl}{\@gls@hyp@opt\@GLSpl}
\glsmfublocker{\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\glscapscase\@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 \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\glscapscase\@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}{\glssentencecase{#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 `\@gls@field@link{⟨options⟩}{⟨label⟩}{⟨link text⟩}`

```
\newcommand{\@gls@field@link}[3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \@gls@link[#1]{#2}{#3}%
  }%

  \glspostlinkhook
}
```

\glstext behaves like \gls except it always uses the value given by the text key and it doesn't mark the entry as used.

\glstext

```
\newrobustcmd*{\glstext}{\@gls@hyp@opt\@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}{\glsentrytext{#2}#3}%
}
```

\GLStext behaves like \glstext 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 \glstext except that the first letter of the text is converted to uppercase.

\Glstext

```
\newrobustcmd*{\Glstext}{\@gls@hyp@opt\@Glstext}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glstext}{\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}{\Glsentryname{#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}{\glsuppercase{\glsentryname{#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}{\glsentrydesc{#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}}%
}
```

\glssymbol behaves like \gls except it always uses the value given by the symbol key and it doesn't mark the entry as used.

\glssymbol

```
\newrobustcmd*{\glssymbol}{\@gls@hyp@opt\@glssymbol}
```

Defined 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 \glssymbol except that the first letter is converted to uppercase.

\Glssymbol

```
\newrobustcmd*{\Glssymbol}{\@gls@hyp@opt\@Glssymbol}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glssymbol}{\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}{\Glsentrysymbol{#2}#3}%
}
```

\GLSsymbol behaves like \glssymbol 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}}%
}
```

\glssymbolplural behaves like \gls except it always uses the value given by the symbolplural key and it doesn't mark the entry as used.

\glssymbolplural

```
\newrobustcmd*{\glssymbolplural}{\@gls@hyp@opt\@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}{\glsentrysymbolplural{#2}#3}%
}
```

\Glssymbolplural behaves like \glssymbolplural except that the first letter is converted to uppercase.

\Glssymbolplural

```
\newrobustcmd*{\Glssymbolplural}{\@gls@hyp@opt\@Glssymbolplural}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glssymbolplural}{\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}{\Glsentrysymbolplural{#2}#3}%
}
```

\GLSsymbolplural behaves like \glssymbolplural 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\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsentryshort{#2}}#3%
    }%
```

Call `\@gls@link` Note that `\@gls@link` sets `\glstype`.

```
    \@gls@link[#1]{#2}{\csname gls@\glstype @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\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
```

Call `\@gls@link` Note that `\@gls@link` sets `\glstype`.

```
    \@gls@link[#1]{#2}{\csname gls@\glstype @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@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}%
    }%
```

Call `\@gls@link` Note that `\@gls@link` sets `\glstype`.

```
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
```

```
    \glspostlinkhook
  }
```

Short plural:

\acrshortpl

```
  \newrobustcmd*{\acrshortpl}{\@gls@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@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

      \def\glslabel{#2}%
      \let\glsifplural\@firstoftwo
      \let\glscapscase\@firstofthree
      \let\glsinsert\@empty
      \def\glscustomtext{%
        \acronymfont{\glsentryshortpl{#2}}#3%
      }%
```

Call \@gls@link Note that \@gls@link sets \glstype.

```
      \@gls@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 `\glstype`.

```
    \@gls@link[#1]{#2}{\csname gls@\glstype @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\glscapscase\@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\glscapscase\@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\glscapscase\@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\glscapscase\@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}[]}%
}
```

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\@thirdofthree
    \let\glsinsert\@empty
```

Bug fix v4.02 removed `\acronymfont` from `\glscustomtext` (`\acronymfont` only designed for short form).

```
    \def\glscustomtext{%
      \glsuppercase{\glsentrylongpl{#2}#3}%
    }%
```

Call `\@gls@link`. Note that `\@gls@link` sets `\glstype`.

```
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%

  \glspostlinkhook
}
```

2.11.2 Displaying entry details without adding information to the glossary

These commands merely display entry information without adding entries in the associated file or having hyperlinks.

`\@gls@entry@field` Generic version.

```
\@gls@entry@field{⟨label⟩}{⟨field⟩}
```

Now using `\csuse` instead of `\csname` to expand to nothing rather than `\relax` if undefined.

```
\newcommand*{\@gls@entry@field}[2]{%
  \csuse{glo@\glsdetoklabel{#1}@#2}%
}
```

`\glsletentryfield`

```
\glsletentryfield{⟨cs⟩}{⟨label⟩}{⟨field⟩}
```

```
\newcommand*{\glsletentryfield}[3]{%
  \letcs{#1}{glo@\glsdetoklabel{#2}@#3}%
}
```

`\@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\glssentencecase\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

\@Glsentryfield{⟨*entry-label*⟩}{⟨*field-label*⟩}

```
\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@acrentryname Now the behaviour when \setacronymstyle is used. The original definition of this command used mfirstuc internal commands that are now being deprecated.

```
\newrobustcmd*{\@Gls@acrentryname}[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\glssentencecase\expandafter{\@glo@text}%
  }%
  {%
```

Not an acronym

```
    \@Gls@entry@field{#1}{name}%
  }%
}
```

\@org@Gls@acrentryname Original definition of \@Gls@acrentryname. This will be removed in a later version.

```
\newcommand*{\@org@Gls@acrentryname}[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\glssentencecase\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]{\@Glsentryfield{#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}
```

\glsentryuserv 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*{\glsentryuserv}[1]{%
  \@gls@entry@field{#1}{userv}%
}
```

\Glsentryuserv

```
\newcommand*{\Glsentryuserv}[1]{%
  \@Glsentryfield{#1}{userv}%
}

\glsmfuexcl{\Glsentryuserv}
```

\glsentryuservi 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*{\glsentryuservi}[1]{%
  \@gls@entry@field{#1}{uservi}%
}
```

\Glsentryuservi

```
\newcommand*{\Glsentryuservi}[1]{%
  \@Glsentryfield{#1}{uservi}%
}

\glsmfuexcl{\Glsentryuservi}
```

\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]{%
  \acrfullformat{\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@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
    }%
  }
}
```

\glsnumlistsep

```
\newcommand*{\glsnumlistsep}{, }
```

\glsnumlistlastsep

```
\newcommand*{\glsnumlistlastsep}{ \& }
```

\glshyperlink Provide a hyperlink to a glossary entry without adding information to the glossary file. The entry needs to be added using a command like \glslink or \glsadd 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*{\glshyperlink}[2][\glsentrytext{\@glo@label}]{%
 \def\@glo@label{#2}%
 \@glslink{\glolinkprefix\glsdetoklabel{#2}}{#1}}
```

2.12 Adding an entry to the glossary without generating text

The following keys are provided for `\glsadd` and `\glsaddall`:

```
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}

\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
```

This key is only used by `\glsaddall`:

```
\define@key{glossadd}{types}{\def\@glo@type{#1}}
```

`\glsadd[`⟨*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 \MFUskippunc. 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, \@glsl@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 glssymbols and glsnumbers, the group titles can be translated (so that \glssymbolsgroupname replaces glssymbols and \glsnumbersgroupname replaces glsnumbers) using the command \glsgetgrouptitle which is defined in glossary-hypernav. This is done to prevent any problem characters in \glssymbolsgroupname 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
```

SetXdyFirstLetterAfterDigits 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]{%
    \glsnoxindywarning\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]{%
    \glsnoxindywarning\GlsSetXdyNumberGroupOrder}
  \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`.

```
\ifglsxindy
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \renewcommand*{\@glsminrange}{#1}}
\else
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
```

```
      \glsnoxindywarning\GlsSetXdyMinRangeLength}
  \fi
```

`\writeist`

```
  \ifglsxindy
```

Code to use if `xindy` is required.

```
    \def\writeist{%
```

Define write register if not already defined

```
      \ifundef{\glswrite}{\newwrite\glswrite}{}%
```

Update attributes list

```
      \@gls@addpredefinedattributes
```

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
      }%
```

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 `{`⟨*Hprefix*⟩`}{`⟨*number*⟩`}`, so need to add all possible combinations of location types.

```
      \@for\@gls@classI:=\@gls@xdy@locationlist\do{%
```

Case where ⟨*Hprefix*⟩ 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"\glspercentchar\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 explicity.

```
\@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
```

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"glssymbols\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@checkescbar####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
```

\@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 TEX is trying to write to a non-existant 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 @glotype@#2@out\endcsname
    \PackageInfo{glossaries}{Writing glossary file
       \jobname.\csname @glotype@#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}}%
       \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{-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
```

`\@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}}%
       \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
```

`\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}%
      {\immediate\write18{\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}{%
 \ifglsxindy
   \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 subsequence 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@dodeflistparser{\noexpand\DeclareListParser
    {\noexpand\glsnumlistparser}{\delimN}}%
  \@gls@dodeflistparser
\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}
```

ssaries@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.

```
  \@glo@check@sortallowed\makenoidxglossaries
```

Implement nonumberlist

```
  \let\@gls@savenonumberlist\gls@noidx@nonumberlist
 }%
}
```

Preamble-only command:

```
\@onlypreamble{\makenoidxglossaries}
```

\glsnumberlistloop

\glsnumberlistloop{⟨*label*⟩}{⟨*handler*⟩}{⟨*xr handler cs*⟩}

```
\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.

```
  \forallglossaries{\@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

```
\@gls@glossary{⟨type⟩}{⟨indexing info⟩}
```

(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\begingroup\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]{%
  \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
}
```

\@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 esclocations=true.

```
\newcommand*{\gls@disablepagerefexpansion}{%
  \glswrglossdisablelocationcmds
  \ifglsesclocations
    \@for\@gls@this:=\gls@protected@pagefmts\do
```

```
    {%
      \expandafter\let\@gls@this\relax
    }%
  \fi
}
```

lswrglossdisablelocationcmds

```
\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

\glsaddprotectedpagefmt{⟨*cs name*⟩}

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 (\⟨*csname*⟩\c@page must be valid).

```
\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
}
```

\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]{%
  \ifglsesclocations
    \@@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{\theglsentrycounter}%
  \expandafter\def\expandafter\@glsHlocref\expandafter{\theHglsentrycounter}%
```

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 \theglsentrycounter and \theHglsentrycounter 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{\theglsentrycounter}%
  \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\theHglsentrycounter\theglsentrycounter\relax
    \def\@glo@counterprefix{}%
  \else
    \protected@edef\@glsHlocref{\theHglsentrycounter}%
    \@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

```
  \ifglsxindy
```

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}{\@glsnumberformat}%
      {\@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}{\@glslocref}}%
  \fi
}
```

\@xp@gls@getcounterprefix

```
\newcommand*\@xp@gls@getcounterprefix[2]{%
 \bgroup
   \glswrglossdisableanchorcmds
   \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^^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
}
```

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 `[`⟨*tag*⟩`]{`⟨*list*⟩`}`, where ⟨*tag*⟩ is a tag such as "see" and ⟨*list*⟩ 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*⟩`}{`⟨*cross-ref list*⟩`}`

```
  \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]{\glshyperlink[\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@\glscurrententrylabel @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}}%
  }%
}
```

oss@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=\glsdefaulttype]{%
  \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. Entries won't be sorted and the location list will be empty.

```
\newcommand*{\printnoidxglossary}[1][type=\glsdefaulttype]{%
  \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 @glotype@\@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@dotoctitle{\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 @glotype@\@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@dotoctitle
```

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 @glotype@\@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 @glotype@\@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\@xdylanguage[2]{}}%
          \noexpand\immediate\noexpand\write\@auxout{%
          \string\@xdylanguage{\@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}%
}%
}
```

TODO: rewrite sorting code (add provide rollback).

The sort macros all have the syntax:

\@glo@sortmacro@⟨*order*⟩{⟨*type*⟩}

where ⟨*order*⟩ is the sort order as specified by the sort key and ⟨*type*⟩ is the glossary type. (The referenced entry list is stored in \@glsref@⟨*type*⟩. The actual sorting is done by \@glo@sortentries{⟨*handler*⟩}{⟨*type*⟩}.

\@glo@sortentries

```
\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?

```
    \xifinlistcs{\@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.

```
      \xifinlistcs{\@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

\@glo@sortedinsert{⟨*list*⟩}{⟨*entry label*⟩}

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{\@glo@sorthandler@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{\@glo@sorthandler@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}%
    {%
      \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]}{}%
    }%
  }
```

`\@glo@sortmacro@case` Sort macro for 'case'

```
\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]}{}%
  }%
}
```

`\@glo@sortmacro@nocase` Sort macro for 'nocase'

```
\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]}{}%
  }%
}
```

`\@glo@sortmacro@def` Sort macro for 'def'. The order of definition is given in `\glolist@`⟨*type*⟩.

```
\newcommand*{\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{}%
  \forglsentries[#1]{\@gls@thislabel}%
  {%
    \xifinlistcs{\@gls@thislabel}{@glsref@#1}%
    {%
      \listeadd{\@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 `\printnoidxglossary` which doesn't use an indexing application. Since `\printnoidxglossary` 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}%
```

Fetch the first letter:

```
  \expandafter\glo@grabfirst\@gls@sort{}{}\@nil
  \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}%
  }%
}
```

lsnoidxdisplayloclisthandler 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

\glsnoidxdisplayloc{⟨*prefix*⟩}{⟨*counter*⟩}{⟨*format*⟩}
{⟨*location*⟩}

Display a location in the location list.

```
\newcommand*\glsnoidxdisplayloc[4]{%
  \setentrycounter[#1]{#2}%
  \csuse{#3}{#4}%
}
```

\@gls@reference

\@gls@reference{⟨*type*⟩}{⟨*label*⟩}{⟨*loc*⟩}

Identifies that a reference has been used (for use in the aux file). All entries must be defined in the preamble.

```
\newcommand*{\@gls@reference}[3]{%
```

Add to label list

```
  \glsdoifexistsorwarn{#2}%
  {%
    \ifcsundef{@glsref@#1}{\csgdef{@glsref@#1}{}}{}%
    \ifinlistcs{#2}{@glsref@#1}%
    {}%
    {\listcsgadd{@glsref@#1}{#2}}%
```

Add to location list

```
    \ifcsundef{glo@\glsdetoklabel{#2}@loclist}%
    {\csgdef{glo@\glsdetoklabel{#2}@loclist}{}}%
    {}%
    \listcsgadd{glo@\glsdetoklabel{#2}@loclist}{#3}%
  }%
}
```

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@dotoctitle\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@dotoctitle\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{glsentrycounter#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 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*{\@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 subcounter 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
}
```

\glsreseteentrycounter Resets the glossaryentry counter.

```
\newcommand*{\glsreseteentrycounter}{%
  \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]{%
 \ifglsentrycounter
   \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
 \else
   \ifglssubentrycounter
     \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
   \else
     \gls{#1}%
   \fi
 \fi
}
```

`\glsentrycounterlabel` Defines how to display the glossaryentry counter.

```
\newcommand*{\glsentrycounterlabel}{%
 \ifglsentrycounter
   \theglossaryentry.\space
 \fi
}
```

`\glssubentrycounterlabel` Defines how to display the glossarysubentry counter.

```
\newcommand*{\glssubentrycounterlabel}{%
 \ifglssubentrycounter
   \theglossarysubentry)\space
 \fi
}
```

`\glsentryitem` Step and display glossaryentry counter, if appropriate.

```
\newcommand*{\glsentryitem}[1]{%
  \ifglsentrycounter
    \glsstepentry{#1}\glsentrycounterlabel
  \else
    \glsresetsubentrycounter
  \fi
}
```

`\glssubentryitem` Step and display glossarysubentry counter, if appropriate.

```
\newcommand*{\glssubentryitem}[1]{%
  \ifglssubentrycounter
    \glsstepsubentry{#1}\glssubentrycounterlabel
  \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{\glolinkprefix#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@\glsdetoklabel{#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}%
    {%
       \glsentrydesc{#1}%
    }%
  }

\Glossentrydesc
  \newcommand*{\Glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \Glsentrydesc{#1}%
    }%
  }
  \glsmfuaddmap{\glossentrydesc}{\Glossentrydesc}

\glossentrysymbol
  \newcommand*{\glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
       \glsentrysymbol{#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: `glssymbols`, `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 LaTeX3 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 \glshypernumber. 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.

\glshypernumber

```
\ifcsundef{hyperlink}%
{%
  \def\glshypernumber#1{#1}%
}%
{%
  \def\glshypernumber#1{\@xp@glshypernumber{#1}}
}
```

\@xp@glshypernumber

```
\newrobustcmd{\@xp@glshypernumber}[1]{%
  \@glshypernumber#1\nohyperpage{}\@nil
}
```

\@glshypernumber This code was provided by Heiko Oberdiek to allow material to be attached to the location.

```
\def\@glshypernumber#1\nohyperpage#2#3\@nil{%
  \ifx\\#1\\%
  \else
    \gls@delimR#1\delimR\delimR\\%
  \fi
  \ifx\\#2\\%
  \else
    #2%
  \fi
  \ifx\\#3\\%
  \else
    \@glshypernumber#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 \glshypernumber).

\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{\glsentrycounter\@glo@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{\glshypernumber{#1}}}
```

\hypersf
```
\newcommand*{\hypersf}[1]{\textsf{\glshypernumber{#1}}}
```

\hypertt
```
\newcommand*{\hypertt}[1]{\texttt{\glshypernumber{#1}}}
```

\hyperbf
```
\newcommand*{\hyperbf}[1]{\textbf{\glshypernumber{#1}}}
```

\hypermd
```
\newcommand*{\hypermd}[1]{\textmd{\glshypernumber{#1}}}
```

\hyperit
```
\newcommand*{\hyperit}[1]{\textit{\glshypernumber{#1}}}
```

\hypersl
```
\newcommand*{\hypersl}[1]{\textsl{\glshypernumber{#1}}}
```

\hyperup
```
\newcommand*{\hyperup}[1]{\textup{\glshypernumber{#1}}}
```

\hypersc
```
\newcommand*{\hypersc}[1]{\textsc{\glshypernumber{#1}}}
```

\hyperemph
```
\newcommand*{\hyperemph}[1]{\emph{\glshypernumber{#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}}%
    }%
  }%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
      \noexpand\gls{#1}\noexpand\xspace}%
    }%
  }%
}
```

\newacronym[⟨*key-val list*⟩]{⟨*label*⟩}{⟨*abbrev*⟩}{⟨*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:

```
    \Acrfullfmt{#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:

```
    \ACRfullfmt{#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]{%
```

Make it easier for acronym styles to change this:

```
  \ACRfullplfmt{#1}{#2}{#3}%
}
```

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
```

\glslongtok

```
\newtoks\glslongtok
```

\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@acrentryname
```

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}{%
      \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}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \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}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \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` ⟨*short*⟩ (⟨*long*⟩) acronym style.

```
\newacronymstyle{short-long}%
{%
```

Check for long form in case this is a mixed glossary.

```
  \ifglshaslong{\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}%
}
```

`long-sc-short` ⟨*long*⟩ (`\textsc{`⟨*short*⟩`}`) 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` ⟨*long*⟩ ({⟨*short*⟩}) 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` ⟨*long*⟩ (`\textsc{`⟨*short*⟩`}`) 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` ⟨*long*⟩ (`\textsmaller{`⟨*short*⟩`}`) 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` ⟨*short*⟩ ({⟨*long*⟩}) 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]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##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]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
```

`sm-short-long-desc` ⟨*long*⟩ (`\textsmaller{`⟨*short*⟩`}`) acronym style that has an accompanying description (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]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##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}{\glsgenacfmt}{\glsgenentryfmt}%
 }%
 {%
   \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}{%
       \glsuppercase{\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}{%
       \glsuppercase{\acronymfont{\glsentryshortpl{##2}}##3\space
       (\glsentrylongpl{##2})}}}%
```

Similarly for `\glsentryfull` etc:

```
   \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})}%
}
```

`footnote-sc` `\textsc{`⟨*short*⟩`}\footnote{`⟨*long*⟩`}` acronym style.

```
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##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{\glsentryshort{##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]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##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\Acrfull
```

First letter upper case plural full form

\Acfp

```
\let\Acfp\Acrfullpl
```

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
```

etDefaultAcronymDisplayStyle Sets the default acronym display style for given glossary. Used by glossaries-extra when restoring \newacronym.

```
\newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsgenentryfmt}%
}
```

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.

xpand@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}}%
}
```

\acrnolinkfootnote This was used by the old footnote acronym styles. Since it may have been used outside of those styles, it has been retained.

```
\newcommand*{\acrnolinkfootnote}[3]{%
  \footnote{#3}%
}
```

nFootnoteAcronymDisplayStyle Sets the acronym display style for given glossary for the description and footnote combination. Deprecated following the introduction of \setacronymstyle. Removed in v4.50.

riptionFootnoteNewAcronymDef Deprecated following the introduction of \setacronymstyle. Removed in v4.50.

criptionFootnoteAcronymStyle If a description and footnote are both required, store the long form in the symbol key. Store the short form in text key. Note that since the long form is stored in the symbol key, if you want the long form to appear in the list of acronyms, you need to use a glossary style that displays the symbol key. This style was deprecated with the introduction of \newacronymstyle. Deprecated following the introduction of \setacronymstyle. Removed in v4.50.

iptionDUAAcronymDisplayStyle Sets the acronym display style for given glossary with description and dua combination. Deprecated following the introduction of \setacronymstyle. Removed in v4.50.

\DescriptionDUANewAcronymDef Deprecated following the introduction of \setacronymstyle. Removed in v4.50.

etDescriptionDUAAcronymStyle Description, don't use acronym and no footnote. Note that the short form is stored in the symbol key, so if the short form needs to be displayed in the glossary, use a style the displays the symbol. This style was deprecated with the introduction of \newacronymstyle. Deprecated following the introduction of \setacronymstyle. Removed in v4.50.

scriptionAcronymDisplayStyle Sets the acronym display style for given glossary using the description setting (but not footnote or dua). Deprecated following the introduction of \setacronymstyle. Removed in v4.50.

\DescriptionNewAcronymDef Deprecated following the introduction of \setacronymstyle. Removed in v4.50.

\SetDescriptionAcronymStyle Option description is used, but not dua or footnote. Store long form in first key and short form in text and symbol key. The name is stored using \acrnameformat to allow the user to override the way the name is displayed in the list of acronyms. Deprecated following the introduction of \setacronymstyle. Removed in v4.50.

tFootnoteAcronymDisplayStyle Sets the acronym display style for given glossary with footnote setting (but not description or dua). Deprecated following the introduction of \setacronymstyle. 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
}
```

\showglotype

\showglotype{⟨*label*⟩}

```
\newcommand*{\showglotype}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@type\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@\glsdetoklabel{#1}@useriii\endcsname
}
```

\showglouseriv

\showglouseriv{⟨label⟩}

```
\newcommand*{\showglouseriv}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@useriv\endcsname
}
```

\showglouserv

\showglouserv{⟨label⟩}

```
\newcommand*{\showglouserv}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@userv\endcsname
}
```

\showglouservi

\showglouservi{⟨label⟩}

```
\newcommand*{\showglouservi}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@uservi\endcsname
}
```

\showgloname

\showgloname{⟨label⟩}

```
\newcommand*{\showgloname}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@name\endcsname
}
```

\showglodesc

\showglodesc{⟨label⟩}

```
\newcommand*{\showglodesc}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#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@#1@in\endcsname
}
```

\showglossaryout

```
\showglossaryout{⟨glossary-label⟩}
```

Show the 'out' extension for the given glossary.

```
\newcommand*{\showglossaryout}[1]{%
  \expandafter\show\csname @glotype@#1@out\endcsname
}
```

\showglossarytitle

```
\showglossarytitle{⟨glossary-label⟩}
```

Show the title for the given glossary.

```
\newcommand*{\showglossarytitle}[1]{%
  \expandafter\show\csname @glotype@#1@title\endcsname
}
```

\showglossarycounter

```
\showglossarycounter{⟨glossary-label⟩}
```

Show the counter for the given glossary.

```
\newcommand*{\showglossarycounter}[1]{%
  \expandafter\show\csname @glotype@#1@counter\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{v4.54}{2024-04-03}
```

Package Definition:

```
\ProvidesPackage{glossary-hypernav}[2024/04/03 v4.54 (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 where 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 }
        }
   }
   {
      \seq_new:c { g_glossaries_ #1 _hypergroups_seq }
      \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

\glsnavhypertarget[⟨*type*⟩]{⟨*label*⟩}{⟨*text*⟩}

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, \@glo@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]{
  \@glstarget{\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 verions 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{v4.54}{2024-04-03}
```

Declare package:

```
\ProvidesPackage{glossary-inline}[2024/04/03 v4.54 (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{v4.54}{2024-04-03}
```

Declare package:

```
\ProvidesPackage{glossary-list}[2024/04/03 v4.54 (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 gettitlestring:

```
\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
    \glssubentryitem{##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 hypertarget:

```
  \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{v4.54}{2024-04-03}
```

Declare package:

```
\ProvidesPackage{glossary-long}[2024/04/03 v4.54 (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]{%
     \glsentryitem{##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|p{\glsdescwidth}|l|p{\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 glostylelong4colheaderborder 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{v4.54}{2024-04-03}
```

Declare package:

```
\ProvidesPackage{glossary-longbooktabs}[2024/04/03 v4.54 (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{\glsdescwidth}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}{l>{\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}{l>{\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}{l>{\raggedright}p{\glsdescwidth}l%
     >{\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\vsize\@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\vsize\@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{v4.54}{2024-04-03}
```

Declare package:

```
\ProvidesPackage{glossary-longragged}[2024/04/03 v4.54 (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}{l>{\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]{%
  \glsentryitem{##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 glostylelongraggedborder 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}{l>{\raggedright}p{\glsdescwidth}l%
     >{\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}{l>{\raggedright}p{\glsdescwidth}l%
    >{\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 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}{|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}%
}
```

ltlongragged4colheaderborder The altlongragged4colheaderborder style is like the above but with a header as well as a border.

```
\newglossarystyle{altlongragged4colheaderborder}{%
```

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}{|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{v4.54}{2024-04-03}
```

Declare package:

```
\ProvidesPackage{glossary-mcols}[2024/04/03 v4.54 (NLCT)]
```

Required packages:

```
\RequirePackage{multicol}
\RequirePackage{glossary-tree}
```

`\indexspace` The 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{\glsgetgrouptitle{##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}{\glsmcols}[\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}{\glsmcols}
     \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}%
}
```

`mcoltreespannav` Similar to the mcoltreehypergroup style but the navigation line is put in the optional argument of the multicols environment.

```
\newglossarystyle{mcoltreespannav}{%
  \setglossarystyle{tree}%
  \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}%
}
```

`mcoltreenoname` Multi-column index style. Same as the treenoname, but puts the glossary in multiple columns.

```
\newglossarystyle{mcoltreenoname}{%
  \setglossarystyle{treenoname}%
  \renewenvironment{theglossary}%
  {%
     \begin{multicols}{\glsmcols}
     \setlength{\parindent}{0pt}%
     \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}
```

`mcoltreenonamegroup` Like the mcoltreenoname style but the glossary groups have headings.

```
\newglossarystyle{mcoltreenonamegroup}{%
```

Base it on the glostylemcoltreenoname style:

```
   \setglossarystyle{mcoltreenoname}%
```

Give each group a heading:

```
   \renewcommand{\glsgroupheading}[1]{\par
     \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
```

`mcoltreenonamehypergroup` The mcoltreenonamehypergroup style is like the mcoltreenonamegroup style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{mcoltreenonamehypergroup}{%
```

Base it on the glostylemcoltreenoname style:

```
   \setglossarystyle{mcoltreenoname}%
```

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}%
}
```

`mcoltreenonamespannav` Similar to the mcoltreenonamehypergroup style but the navigation line is put in the optional argument of the multicols environment.

```
\newglossarystyle{mcoltreenonamespannav}{%
  \setglossarystyle{treenoname}%
  \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}%
}
```

`mcolalttree` Multi-column index style. Same as the alttree, but puts the glossary in multiple columns.

```
\newglossarystyle{mcolalttree}{%
  \setglossarystyle{alttree}%
  \renewenvironment{theglossary}%
  {%
     \begin{multicols}{\glsmcols}
     \def\@gls@prevlevel{-1}%
     \mbox{}\par
  }%
  {\par\end{multicols}}%
}
```

`mcolalttreegroup` Like the mcolalttree style but the glossary groups have headings.

```
\newglossarystyle{mcolalttreegroup}{%
```

Base it on the glostylemcolalttree style:

```
  \setglossarystyle{mcolalttree}%
```

Give each group a heading.

```
  \renewcommand{\glsgroupheading}[1]{\par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
```

`mcolalttreehypergroup` The mcolalttreehypergroup style is like the mcolalttreegroup style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{mcolalttreehypergroup}{%
```

Base it on the glostylemcolalttree style:

```
  \setglossarystyle{mcolalttree}%
```

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}{\glsmcols}[\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{v4.54}{2024-04-03}
```

Declare package:

```
\ProvidesPackage{glossary-super}[2024/04/03 v4.54 (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*{\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
}
```

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}{lp{\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}{llll}}{%
     \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}{llll}}%
  {\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}{|l|l|l|l|}}%
  {\end{supertabular}}%
}
```

super4colheaderborder The super4colheaderborder style is like the super4col but with a header and border.

```
\newglossarystyle{super4colheaderborder}{%
```

Base it on the glostylesuper4col 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}{|l|l|l|l|}}%
  {\end{supertabular}}%
}
```

altsuper4col The altsuper4col glossary style is like super4col but has provision for multiline descriptions.

```
\newglossarystyle{altsuper4col}{%
```

Base it on the glostylesuper4col 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 glostylesuper4colheader 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}%
       {|l|p{\glsdescwidth}|l|p{\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}%
       {|l|p{\glsdescwidth}|l|p{\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{v4.54}{2024-04-03}
```

Declare package:

```
\ProvidesPackage{glossary-superragged}[2024/04/03 v4.54 (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}{l>{\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}}%
}
```

rraggedright3colheaderborder 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)

```
  \ifglsnogroupskip
    \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\tabularnewline}\tabletail{}%
     \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}l%
       >{\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}%
       {|l|>{\raggedright}p{\glsdescwidth}|l|%
         >{\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}%
     {|l|>{\raggedright}p{\glsdescwidth}|l|%
        >{\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{v4.54}{2024-04-03}
```

Declare package:

```
\ProvidesPackage{glossary-tree}[2024/04/03 v4.54 (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 alttree style).

```
\newcommand{\glstreepredesc}{\space}
```

\glstreechildpredesc Allow the user to adjust the space before the description for sub-entries (except for the treenoname and alttree 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\glsentryitem{##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*{\glsgroupheading}[1]{%
  \par\noindent
  \glstreegroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}
```

\glsfindwidesttoplevelname Find the widest name over all parentless entries in the given glossary or glossaries.

```
\newrobustcmd*{\glsfindwidesttoplevelname}[1][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@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
      }%
    }%
  }%
}
```

\glssetwidest `\glssetwidest[`⟨*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*{\glssetwidest}[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 `\glssetwidest`.

```
\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[0pt][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}{\ifglsnogroupskip\else\indexspace\fi}%
}
```

alttreegroup Like the alttree style but the glossary groups have headings.

```
\newglossarystyle{alttreegroup}{%
```

Base it on the glostylealttree style:

```
  \setglossarystyle{alttree}%
```

Give each group a heading.

```
  \renewcommand{\glsgroupheading}[1]{\par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
    \par\indexspace}%
}
```

alttreehypergroup The alttreehypergroup style is like the alttreegroup style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{alttreehypergroup}{%
```

Base it on the glostylealttree style:

```
  \setglossarystyle{alttree}%
```

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{v4.54}{2024-04-03}
```

Declare package

```
\ProvidesPackage{glossaries-babel}[2024/04/03 v4.54 (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{v4.54}{2024-04-03}
```

Declare package

```
\ProvidesPackage{glossaries-polyglossia}[2024/04/03 v4.54 (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{v4.54}{2024-04-03}
```

Declare package:

```
\ProvidesPackage{glossaries-prefix}[2024/04/03 v4.54 (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{\glsentryprefixfirst}{\Glsentryprefixfirst}
```

`\Glsentryprefixfirstplural`

```
\newcommand*{\Glsentryprefixfirstplural}[1]{%
  \@Glsentryfield{#1}{prefixfirstplural}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsentryprefixfirstplural}{\Glsentryprefixfirstplural}
```

`\Glsentryprefix`

```
\newcommand*{\Glsentryprefix}[1]{%
 \@Glsentryfield{#1}{prefix}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsentryprefix}{\Glsentryprefix}
```

`\Glsentryprefixplural`

```
\newcommand*{\Glsentryprefixplural}[1]{%
  \@Glsentryfield{#1}{prefixplural}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsentryprefixplural}{\Glsentryprefixplural}
```

Define commands to determine if the prefix keys have been set:

`\ifglshasprefix`

```
\newcommand*{\ifglshasprefix}[3]{%
  \ifcsempty{glo@\glsdetoklabel{#1}@prefix}%
  {#3}%
  {#2}%
}
```

`\ifglshasprefixplural`

```
\newcommand*{\ifglshasprefixplural}[3]{%
  \ifcsempty{glo@\glsdetoklabel{#1}@prefixplural}%
  {#3}%
  {#2}%
}
```

`\ifglshasprefixfirst`

```
\newcommand*{\ifglshasprefixfirst}[3]{%
  \ifcsempty{glo@\glsdetoklabel{#1}@prefixfirst}%
  {#3}%
  {#2}%
}
```

\ifglshasprefixfirstplural

```
\newcommand*{\ifglshasprefixfirstplural}[3]{%
  \ifcsempty{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:

\pglspl

```
\newrobustcmd{\pglspl}{\@gls@hyp@opt\@pglspl}
```

\@pglspl Unstarred version.

```
\newcommand*{\@pglspl}[2][]{%
  \new@ifnextchar[%
  {\@pglspl@{#1}{#2}}%
  {\@pglspl@{#1}{#2}[]}%
}
```

\@pglspl@ Read in the final optional argument:

```
\def\@pglspl@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglshasprefixplural{#2}{\glsentryprefixplural{#2}\glsprefixsep}{}%
    }%
    {%
      \ifglshasprefixfirstplural{#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}%
      {%
        \Glsentryprefix{#2}%
        \glsprefixsep
        \@gls@{#1}{#2}[#3]%
      }%
      {\@Gls@{#1}{#2}[#3]}%
    }%
    {%
      \ifglshasprefixfirst{#2}%
      {%
        \Glsentryprefixfirst{#2}%
        \glsprefixsep
        \@gls@{#1}{#2}[#3]%
      }%
      {\@Gls@{#1}{#2}[#3]}%
    }%
  }%
}
```

Similarly for the plural version:

`\Pglspl`

```
\newrobustcmd{\Pglspl}{\@gls@hyp@opt\@Pglspl}
```

Add sentence case mapping.

```
\glsmfuaddmap{\pglspl}{\Pglspl}
```

`\@Pglspl` Unstarred version.

```
\newcommand*{\@Pglspl}[2][]{%
  \new@ifnextchar[%
  {\@Pglspl@{#1}{#2}}%
  {\@Pglspl@{#1}{#2}[]}%
}
```

`\@Pglspl@` Read in the final optional argument:

```
\def\@Pglspl@#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]}%
    }%
  }%
}
```

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}
```

\@PGLSpl Unstarred version.

```
\newcommand*{\@PGLSpl}[2][]{%
  \new@ifnextchar[%
  {\@PGLSpl@{#1}{#2}}%
  {\@PGLSpl@{#1}{#2}[]}%
}
```

\@PGLSpl@ Read in the final optional argument:

```
\def\@PGLSpl@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglshasprefixplural{#2}%
        {\glsuppercase{\glsentryprefixplural{#2}\glsprefixsep}}{}%
    }%
    {%
      \ifglshasprefixfirstplural{#2}%
      {\glsuppercase{\glsentryprefixfirstplural{#2}\glsprefixsep}}{}%
    }%
    \@GLSpl@{#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{v4.54}{2024-04-03}
```

Package version number now in line with main glossaries package number.

```
\ProvidesPackage{glossaries-accsupp}[2024/04/03 v4.54 (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
\compatiblesubglossentry

Override style compatibility macros:

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

```
\gls@accessibility{⟨options⟩}{⟨PDF element⟩}{⟨value⟩}
{⟨content⟩}
```

```
\providecommand{\gls@accessibility}[4]{#4}
\ifdefstring\gls@accsupp@engine{accsupp}
{
  \RequirePackage{accsupp}
  \renewcommand{\gls@accessibility}[4]{%
    \BeginAccSupp{#1,#2={#3}}#4\EndAccSupp{}%
  }
}
{}
```

\glsaccessibility

```
\glsaccessibility[⟨options⟩]{⟨PDF element⟩}{⟨value⟩}
{⟨content⟩}
```

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@useriiaccess{\relax}%
  \def\@glo@useriiiaccess{\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}%
}
```

\glsentrylongaccess Get the value of the longaccess key for the entry with the given label:

```
\newcommand*{\glsentrylongaccess}[1]{%
  \@gls@entry@field{#1}{longaccess}%
}
```

\glsentrylongpluralaccess Get the value of the longpluralaccess key for the entry with the given label:

```
\newcommand*{\glsentrylongpluralaccess}[1]{%
  \@gls@entry@field{#1}{longpluralaccess}%
}
```

\glsentryuseriaccess Get the value of the user1access key for the entry with the given label:

```
\newcommand*{\glsentryuseriaccess}[1]{%
  \@gls@entry@field{#1}{useriaccess}%
}
```

\glsentryuseriiaccess Get the value of the user2access key for the entry with the given label:

```
\newcommand*{\glsentryuseriiaccess}[1]{%
  \@gls@entry@field{#1}{useriiaccess}%
}
```

\glsentryuseriiiaccess Get the value of the user3access key for the entry with the given label:

```
\newcommand*{\glsentryuseriiiaccess}[1]{%
  \@gls@entry@field{#1}{useriiiaccess}%
}
```

\glsentryuserivaccess Get the value of the user4access key for the entry with the given label:

```
\newcommand*{\glsentryuserivaccess}[1]{%
  \@gls@entry@field{#1}{userivaccess}%
}
```

\glsentryuservaccess Get the value of the user5access key for the entry with the given label:

```
\newcommand*{\glsentryuservaccess}[1]{%
  \@gls@entry@field{#1}{uservaccess}%
}
```

\glsentryuserviaccess Get the value of the user6access key for the entry with the given label:

```
\newcommand*{\glsentryuserviaccess}[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

```
\glsfieldaccsupp{⟨replacement⟩}{⟨content⟩}{⟨field⟩}{⟨label⟩}
```

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

```
\xglsfieldaccsupp{⟨replacement⟩}{⟨content⟩}{⟨field⟩}
{⟨label⟩}
```

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@\glsdetoklabel{#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}}}%
}
```

scriptionpluralaccessdisplay 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@\glsdetoklabel{#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@\glsdetoklabel{#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@\glsdetoklabel{#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@\glsdetoklabel{#2}@userviaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{userv}{#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.

\glsgenentryfmt Redefine to use accessibility information.

```
\renewcommand*{\glsgenentryfmt}{%
  \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

```
            \glspluralaccessdisplay
               {\glsuppercase{\glsentryplural{\glslabel}}}%
               {\glslabel}%
            \glsuppercase{\glsinsert}%
         }%
         {%
```

First use

```
           \glsfirstpluralacessdisplay
              {\glsuppercase{\glsentryfirstplural{\glslabel}}}%
              {\glslabel}%
           \glsuppercase{\glsinsert}%
         }%
       }%
     }%
     {%
```

Singular form

```
       \glscapscase
       {%
```

Don't adjust case

```
         \ifglsused\glslabel
         {%
```

Subsequent use

```
           \glstextaccessdisplay{\glsentrytext{\glslabel}}{\glslabel}%
           \glsinsert
         }%
         {%
```

First use

```
           \glsfirstaccessdisplay{\glsentryfirst{\glslabel}}{\glslabel}%
           \glsinsert
         }%
       }%
       {%
```

Make first letter upper case

```
         \ifglsused\glslabel
         {%
```

Subsequent use

```
            \glstextaccessdisplay{\Glsentrytext{\glslabel}}{\glslabel}%
            \glsinsert
         }%
         {%
```

First use

```
           \glsfirstaccessdisplay{\Glsentryfirst{\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
    }%
  }
```

`\glsgenacfmt` Redefine to include accessibility information.

```
  \renewcommand*{\glsgenacfmt}{%
    \ifdefempty\glscustomtext
    {%
      \ifglsused\glslabel
      {%
```

Subsequent use:

```
        \glsifplural
        {%
```

Subsequent plural form:

```
          \glscapscase
          {%
```

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

```
          \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. (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\glscapscase\@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\glscapscase\@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
```

```
      \let\glsifplural\@secondoftwo
      \let\glscapscase\@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

      \let\glsifplural\@secondoftwo
      \let\glscapscase\@firstofthree
      \let\glsinsert\@empty
      \def\glscustomtext{%
        \acronymfont{\glslongaccessdisplay{%
          \glsuppercase{\glsentrylong{#2}}}{#2}#3}%
      }%
```

Call \@gls@link

```
      \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
    }%

    \glspostlinkhook
  }
```

\@glstext@

```
  \def\@glstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\glsentrytext{#2}}{#2}#3}%
  }
```

\@Glstext@

```
  \def\@Glstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\Glsentrytext{#2}}{#2}#3}%
  }
```

\@GLStext@

```
  \def\@GLStext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}%
    {\glstextaccessdisplay{\glsuppercase{\glsentrytext{#2}}}{#2}%
     \glsuppercase{#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{\Glsentryfirst{#2}}{#2}#3}%
                }

\@GLSfirst@
                \def\@GLSfirst@#1#2[#3]{%
                  \@gls@field@link{#1}{#2}%
                  {\glsfirstaccessdisplay{\glsuppercase{\glsentryfirst{#2}}}{#2}%
                   \glsuppercase{#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{\glsuppercase{\glsentryplural{#2}}}{#2}%
                   \glsuppercase{#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{\glsuppercase{\glsentryfirstplural{#2}}}{#2}%
                    \glsuppercase{#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}%
                }

\@Glssymbol@
                \def\@Glssymbol@#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{\glsuppercase{\glsentrysymbol{#2}}}{#2}%
                     \glsuppercase{#3}}%
                }

\@glssymbolplural@
                \def\@glssymbolplural@#1#2[#3]{%
                  \@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\glsentrysymbolplural{#2}}{#2}#3}%
                }

\@Glssymbolplural@
                \def\@Glssymbolplural@#1#2[#3]{%
                  \@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\Glsentrysymbolplural{#2}}{#2}#3}%
                }

\@GLSsymbolplural@
                \def\@GLSsymbolplural@#1#2[#3]{%
                  \@gls@field@link{#1}{#2}%
                    {\glssymbolpluralaccessdisplay{\glsuppercase{\glsentrysymbolplural{#2}}}{#2}%
                     \glsuppercase{#3}}%
                }

 \@glsuseri@
                \def\@glsuseri@#1#2[#3]{%
                  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glsentryuseri{#2}}{#2}#3}%
                }

 \@Glsuseri@
                \def\@Glsuser@i#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{\glsuppercase{\glsentryuseri{#2}}}{#2}%
                     \glsuppercase{#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{\glsuppercase{\glsentryuserii{#2}}}{#2}%
                     \glsuppercase{#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{\glsuppercase{\glsentryuseriii{#2}}}{#2}%
                     \glsuppercase{#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{\glsuppercase{\glsentryuseriv{#2}}}{#2}%
                     \glsuppercase{#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` ⟨*long*⟩ (⟨*short*⟩) acronym style.

```
\renewacronymstyle{long-short}%
{%
```

Check for long form in case this is a mixed glossary.

```
  \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}%
}
```

`short-long` ⟨*short*⟩ (⟨*long*⟩) acronym style.

```
\renewacronymstyle{short-long}%
{%
```

Check for long form in case this is a mixed glossary.

```
  \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}%
}
```

`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{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##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 description (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
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
```

Plural form:

```
        \glscapscase
        {%
```

Plural form, don't adjust case:

```
          \glslongpluralaccessdisplay{\glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
```

Plural form, make first letter upper case:

```
          \glslongpluralaccessdisplay{\Glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
```

Plural form, all caps:

```
          \glslongpluralaccessdisplay
            {\glsuppercase{\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
         {\glsuppercase{\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}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \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
         {\glsuppercase
           {\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
        {\glsuppercase
           {\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}{\glstextup{\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

\glsdefaultshortaccess{⟨*long*⟩}{⟨*short*⟩}

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.

riptionFootnoteNewAcronymDef 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

```
      \showglonameaccess
                        \newcommand*{\showglonameaccess}[1]{%
                          \expandafter\show\csname glo@\glsdetoklabel{#1}@access\endcsname
                        }

      \showglotextaccess
                        \newcommand*{\showglotextaccess}[1]{%
                          \expandafter\show\csname glo@\glsdetoklabel{#1}@textaccess\endcsname
                        }

    \showglopluralaccess
                        \newcommand*{\showglopluralaccess}[1]{%
                          \expandafter\show\csname glo@\glsdetoklabel{#1}@pluralaccess\endcsname
                        }

     \showglofirstaccess
                        \newcommand*{\showglofirstaccess}[1]{%
                          \expandafter\show\csname glo@\glsdetoklabel{#1}@firstaccess\endcsname
                        }

\showglofirstpluralaccess
                        \newcommand*{\showglofirstpluralaccess}[1]{%
                          \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpluralaccess\endcsname
                        }

     \showglosymbolaccess
                        \newcommand*{\showglosymbolaccess}[1]{%
                          \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolaccess\endcsname
                        }

\showglosymbolpluralaccess
                        \newcommand*{\showglosymbolpluralaccess}[1]{%
                          \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolpluralaccess\endcsname
                        }

      \showglodescaccess
                        \newcommand*{\showglodescaccess}[1]{%
                          \expandafter\show\csname glo@\glsdetoklabel{#1}@descaccess\endcsname
                        }

 \showglodescpluralaccess
                        \newcommand*{\showglodescpluralaccess}[1]{%
                          \expandafter\show\csname glo@\glsdetoklabel{#1}@descpluralaccess\endcsname
                        }
```

```
\showgloshortaccess
                        \newcommand*{\showgloshortaccess}[1]{%
                          \expandafter\show\csname glo@\glsdetoklabel{#1}@shortaccess\endcsname
                        }

\showgloshortpluralaccess
                        \newcommand*{\showgloshortpluralaccess}[1]{%
                          \expandafter\show\csname glo@\glsdetoklabel{#1}@shortpluralaccess\endcsname
                        }

 \showglolongaccess
                        \newcommand*{\showglolongaccess}[1]{%
                          \expandafter\show\csname glo@\glsdetoklabel{#1}@longaccess\endcsname
                        }

 \showglolongpluralaccess
                        \newcommand*{\showglolongpluralaccess}[1]{%
                          \expandafter\show\csname glo@\glsdetoklabel{#1}@longpluralaccess\endcsname
                        }
```

7 Backwards Compatibility

7.1 glossaries-compatible-207

Provides compatibility with version 2.07 and below. This uses original glossaries xindy 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{v4.54}{2024-04-03}
```

Identify package.

```
\ProvidesPackage{glossaries-compatible-207}[2024/04/03 v4.54 (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{v4.54}{2024-04-03}
```

Identify package.

```
\ProvidesPackage{glossaries-compatible-307}[2024/04/03 v4.54 (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
}%
{%
  \@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
   \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]{%
  \ifstrempty{#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]{}
\glssavenumberlistfalse
\newcommand*\@glo@seeautonumberlist{}
\@gls@declareoption{seeautonumberlist}{%
   \renewcommand*{\@glo@seeautonumberlist}{%
      \def\@glo@prefix{\glsnextpages}%
   }%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{}
\glsesclocationstrue
\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*{\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
      \ifglsentrycounter
        \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*\glssortnumberfmt[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{%
```

```
    \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@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*{\@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@glsisacronymlist
\newcommand*{\gls@checkisacronymlist}[1]{%
   \glsIfListOfAcronyms{#1}%
     {\@glsisacronymlisttrue}{\@glsisacronymlistfalse}%
}
\newcommand*{\SetAcronymLists}[1]{%
  \renewcommand*{\@glsacronymlists}{#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 `theglossary' 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}{%
  \ifglssanitizesort
    \@@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}%
```

```
    \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
    }%
    {}%
  }
}
\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\@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
  \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}
  \@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}%
  \ifglsxindy
    \def\istfilename{\gls@istfilebase.xdy}
  \else
    \def\istfilename{\gls@istfilebase.ist}
  \fi
}
\@onlypremakeg\setStyleFile
\ifglsxindy
  \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]{%
    \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]{\glshypernumber{#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}%
  }%
  \glsglossarymark{\glossarytoctitle}%
}
\ifcsundef{glossarymark}%
{%
  \newcommand{\glsglossarymark}[1]{\glossarymark{#1}}
}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand{\glsglossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUChead{#1}}{\memUChead{#1}}%
      \else
        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand{\glsglossarymark}[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@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]{%
    \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@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{}
\ifglsxindy
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
  \edef\@xdyuseralphabets{%
    \@xdyuseralphabets ^^J
    (define-alphabet "#1" (#2))}}
\else
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
     \glsnoxindywarning\GlsAddXdyAlphabet}
\fi
\ifglsxindy
  \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"\@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]{%
     \glsnoxindywarning\GlsAddXdyLocation}
\fi
\ifglsxindy
  \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
\ifglsxindy
  \newcommand*\GlsSetXdyLocationClassOrder[1]{%
    \def\@xdylocationclassorder{#1}}
\else
  \newcommand*\GlsSetXdyLocationClassOrder[1]{%
    \glsnoxindywarning\GlsSetXdyLocationClassOrder}
\fi
\ifglsxindy
  \def\@xdysortrules{}
\fi
\ifglsxindy
  \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]{%
    \glsnoxindywarning\GlsAddSortRule}
\fi
\ifglsxindy
  \def\@xdyrequiredstyles{tex}
\fi
\ifglsxindy
  \newcommand*\GlsAddXdyStyle[1]{%
    \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
  \newcommand*\GlsAddXdyStyle[1]{%
    \glsnoxindywarning\GlsAddXdyStyle}
\fi
\ifglsxindy
  \newcommand*\GlsSetXdyStyles[1]{%
    \edef\@xdyrequiredstyles{#1}}
\else
  \newcommand*\GlsSetXdyStyles[1]{%
    \glsnoxindywarning\GlsSetXdyStyles}
\fi
\newcommand*{\findrootlanguage}{}
\def\@xdylanguage#1#2{}
\ifglsxindy
  \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{}
\ifglsxindy
  \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*{\forglsentries}[3][\glsdefaulttype]{%
  \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@}%
  {%
    \forglsentries[\@@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}%
     \edef\@gls@thislabel{\glsdetoklabel{#1}}%
     \expandafter\forglsentries\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}%
  {%
    \ifcsempty{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
  }%
}
\newcommand*{\ifglshasdesc}[3]{%
  \ifcsempty{glo@\glsdetoklabel{#1}@desc}%
  {#3}%
  {#2}%
}
\newcommand*{\ifglsdescsuppressed}[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@\glsdetoklabel{#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@\glsdetoklabel{#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*{\defglsentryfmt}[2][\glsdefaulttype]{%
  \csgdef{gls@#1@entryfmt}{#2}%
}
\newcommand*{\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*{\@gls@forbidtexext}[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 @glotype@#2@log\endcsname{#1}%
  \expandafter\edef\csname @glotype@#2@in\endcsname{#3}%
  \expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
  \expandafter\@gls@forbidtexext\csname @glotype@#2@log\endcsname
  \expandafter\@gls@forbidtexext\csname @glotype@#2@in\endcsname
  \expandafter\@gls@forbidtexext\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
  {%
    \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{\glsnonextpages}%
    \@gls@savenonumberlist{true}%
  \else
    \def\@glo@prefix{\glsnextpages}%
    \@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*{\@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]{%
  ??%
  \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*{\glsexpandfields}{%
  \let\gls@assign@field\@gls@expand@fields
}
\newcommand*{\glsnoexpandfields}{%
  \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]{%
  \ifglsentryexists{#1}%
  {}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\@onlypreamble{\provideglossaryentry}
\newrobustcmd{\new@glossaryentry}[2]{%
  \ifundef\@gls@deffile
  {%
     \global\newwrite\@gls@deffile
     \immediate\openout\@gls@deffile=\jobname.glsdefs
  }%
  {}%
  \ifglsentryexists{#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\ifglsentryexists{#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
      }%
    }%
    {}%
  }%
  \glswritedefhook
  \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{\glsfieldxdef}[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{\glsfieldedef}[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{\glsfieldgdef}[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*{\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
  \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}%
\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\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
  \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{@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\@gloname\@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\@glodesc\@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\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]{%
  \edef\@glo@thislabel{\glsdetoklabel{#1}}%
  \edef\glo@type{\csname glo@\@glo@thislabel @type\endcsname}%
  \def\glo@list{,}%
  \forglsentries[\glo@type]{\glo@label}%
   {%
     \ifdefequal\@glo@thislabel\glo@label
       {}{\eappto\glo@list{\glo@label,}}%
   }%
  \cslet{glolist@\glo@type}{\glo@list}%
  \csdef{glo@\@glo@thislabel @type}{#2}%
}
\ifglsxindy
  \newcommand*{\@glossaryentryfield}{\string\\glossentry}
\else
  \newcommand*{\@glossaryentryfield}{\string\glossentry}
\fi
\ifglsxindy
  \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}%
  \ifglsxindy
    \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\@cgls@##1##2[##3]{%
   \ifnum\glsentryprevcount{##2}=1\relax
     \cglsformat{##2}{##3}%
     \glsunset{##2}%
   \else
     \@gls@{##1}{##2}[##3]%
   \fi
 }%
 \def\@cglspl@##1##2[##3]{%
   \ifnum\glsentryprevcount{##2}=1\relax
     \cglsplformat{##2}{##3}%
     \glsunset{##2}%
   \else
     \@glspl@{##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\@cGlspl@##1##2[##3]{%
   \ifnum\glsentryprevcount{##2}=1\relax
     \cGlsplformat{##2}{##3}%
     \glsunset{##2}%
   \else
     \@Glspl@{##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}%
 \@gls@{#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}%
 \@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}
\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}{%
  \@@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
```

```
      {%
        \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
        {%
          \acronymfont{\glsentryshort{\glslabel}}\glsinsert
        }%
        {%
          \acronymfont{\Glsentryshort{\glslabel}}\glsinsert
        }%
        {%
          \mfirstucMakeUppercase
            {\acronymfont{\glsentryshort{\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
  }%
}
```

```
\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}}%
```

```
      }%
      {%
        \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
      \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
    \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\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@dosubst{%
      \noexpand\DTLsubstituteall\noexpand\gls@xdystring
      {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
    }%
    \gls@dosubst
  }%
  \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\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@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{%
```

```
 \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]{%
  \@glsshowtarget{#1}%
  \settoheight{\gls@tmplen}{#2}%
  \raisebox{\gls@tmplen}{\hypertarget{#1}{}}#2%
}
\newcommand*{\glsdohyperlink}[2]{%
 \@glsshowtarget{#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\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*{\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\glscapscase\@secondofthree
    \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@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \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\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\glscapscase\@secondofthree
    \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\glscapscase\@thirdofthree
    \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*{\glsdisp}{\@gls@hyp@opt\@glsdisp}
\newcommand*{\@glsdisp}[3][]{%
  \glsdoifexists{#2}{%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@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*{\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*{\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*{\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*{\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*{\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*{\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*{\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\glscapscase\@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\glscapscase\@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{\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\@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{\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}
\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\@secondofthree
    \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}
\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\glscapscase\@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@acrentryname}[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
  }%
  {%
    \@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*{\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
         \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*{\glshyperlink}[2][\glsentrytext{\@glo@label}]{%
 \def\@glo@label{#2}%
 \@glslink{\glolinkprefix\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\glstildechar{\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]{%
    \glsnoxindywarning\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]{%
    \glsnoxindywarning\GlsSetXdyNumberGroupOrder}
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
    \@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*{\@glsminrange}{2}
\ifglsxindy
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \renewcommand*{\@glsminrange}{#1}}
\else
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \glsnoxindywarning\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@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
      \\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"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
    \@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#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@checkescbar####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@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
}
\ifglsxindy
 \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{@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@dodeflistparser{\noexpand\DeclareListParser
      {\noexpand\glsnumlistparser}{\delimN}}%
    \@gls@dodeflistparser
  \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@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]{%
    \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
}%
}
\@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*{\@glswritefiles}{%
  \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\begingroup\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]{%
  \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}{%
  \@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]{%
  \eappto\gls@protected@pagefmts{,\expandonce{\csname gls#1page\endcsname}}%
  \csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
  \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]{%
  \ifglsesclocations
    \@@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{\theHglsentrycounter}%
  \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\@glslocref{\theglsentrycounter}%
  \endgroup
  \@gls@checkmkidxchars\@glslocref
  \expandafter\ifx\theHglsentrycounter\theglsentrycounter\relax
    \def\@glo@counterprefix{}%
  \else
    \protected@edef\@glsHlocref{\theHglsentrycounter}%
    \@gls@checkmkidxchars\@glsHlocref
    \edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
      {\@glslocref}{\@glsHlocref}%
    }%
    \@do@gls@getcounterprefix
  \fi
  \edef\@gls@label{\glsdetoklabel{#1}}%
  \@@do@@wrglossary
```

```
}
\newcommand*{\@@do@@wrglossary}{%
  \ifglsxindy
    \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
    \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^^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
      :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}%
    \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}{%
  \forallglossaries{\@@glo@type}{\printnoidxglossary[type=\@@glo@type]}%
}
\newcommand*{\@printgloss@setsort}{}
\newcommand*{\@gls@preglossaryhook}{}
\newcommand{\@printglossary}[2]{%
```

```
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{\csname @glotype@\@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@dotoctitle{\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 @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@dotoctitle
    \@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 @glotype@\@glo@type @in\endcsname}%
  \IfFileExists{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
  {}%
  {\null}%
  \ifglsxindy
    \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@\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{}%
  \forglsentries[#1]{\@gls@thislabel}%
  {%
    \xifinlistcs{\@gls@thislabel}{@glsref@#1}%
    {%
      \listeadd{\@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@\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@dotoctitle\relax
}
\define@key{printgloss}{toctitle}{%
 \def\glossarytoctitle{#1}%
 \let\gls@dotoctitle\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{glssubentrycounter#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}{%
  \ifglsentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*{\glsstepentry}[1]{%
  \ifglsentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
\newcommand*{\glsstepsubentry}[1]{%
  \ifglssubentrycounter
    \edef\currentglssubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
  \fi
}
\newcommand*{\glsrefentry}[1]{%
 \ifglsentrycounter
   \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
 \else
   \ifglssubentrycounter
     \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
   \else
     \gls{#1}%
   \fi
 \fi
}
\newcommand*{\glsentrycounterlabel}{%
 \ifglsentrycounter
   \theglossaryentry.\space
 \fi
}
\newcommand*{\glssubentrycounterlabel}{%
```

```
 \ifglssubentrycounter
   \theglossarysubentry)\space
 \fi
}
\newcommand*{\glsentryitem}[1]{%
  \ifglsentrycounter
    \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*{\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]{%
  \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}%
  {%
     \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}{glssymbols}}
            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}{\glssymbolsgroupname}}{glssymbols}{%
\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\csuse{@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\glshypernumber#1{#1}%
```

```
}%
{%
  \def\glshypernumber#1{\@glshypernumber#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\x{\the\toks@}%
   \ifx\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\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\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}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \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}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \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}{\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]{%
    \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
        }%
        {%
          \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}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \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@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:=\@glsacronymlists\do{%
    \SetDefaultAcronymDisplayStyle{\@gls@type}%
  }%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@glsacronymlists\@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}{\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}%
        \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{\glsgenentryfmt}\gls@org@insert
      }%
      {%
        \glsgenentryfmt
        \ifglshassymbol{\glslabel}%
          {%
             \glsifplural
             {%
               \def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%
             }%
             {%
               \def\@glo@symbol{\glsentrysymbol{\glslabel}}%
             }%
             \space(\protect\firstacronymfont
             {\glscapscase
              {\@glo@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
```

```
    \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
        \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]{%
  \defglsentryfmt[#1]{%
    \ifdefempty\glscustomtext
    {%
      \let\gls@org@insert\glsinsert
      \let\glsinsert\@empty
      \ifglsused{\glslabel}%
      {%
        \acronymfont{\glsgenentryfmt}\gls@org@insert
      }%
      {%
        \glsgenentryfmt
        \ifglshassymbol{\glslabel}%
        {%
          \glsifplural
          {%
            \def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%
          }%
          {%
            \def\@glo@symbol{\glsentrysymbol{\glslabel}}%
          }%
          \space
            (\glscapscase
            {\firstacronymfont{\@glo@symbol}}%
            {\firstacronymfont{\@glo@symbol}}%
            {\firstacronymfont{\mfirstucMakeUppercase{\@glo@symbol}}})%
        }%
        {}%
      }%
    }%
    {\glscustomtext\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]{\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@gls@assign@firstpl\gls@assign@firstpl
  \let\@org@gls@assign@plural\gls@assign@plural
  \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
```

```
  \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@symbolplural##1##2{%
    \@@gls@expand@field{##1}{symbolplural}{##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
  \let\gls@assign@descplural\@org@gls@assign@descplural
}
\newcommand*{\SetDUAStyle}{%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@glsacronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetDUADisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \DUANewAcronymDef
  }%
  \@for\@gls@type:=\@glsacronymlists\do{%
    \SetDUADisplayStyle{\@gls@type}%
  }%
}
\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
          \SetDUAStyle
        \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\@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*{\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*{\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@\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 @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 glolist@#1\endcsname
}
\csname ifglscompatible-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 @uservaccess\endcsname{%
      \@glo@uservaccess}%
  \expandafter
    \protected@xdef\csname glo@\@glo@label @useriaccess\endcsname{%
      \@glo@useriaccess}%
}
\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*{\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*{\glstextaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@textaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{text}{#1}{\glsentrytextaccess{#2}}}%
}
\newrobustcmd*{\glspluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@pluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{plural}{#1}{\glsentrypluralaccess{#2}}}%
}
\newrobustcmd*{\glsfirstaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@firstaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{first}{#1}{\glsentryfirstaccess{#2}}}%
}
\newrobustcmd*{\glsfirstpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@firstpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{firstpl}{#1}{\glsentryfirstpluralaccess{#2}}}%
}
\newrobustcmd*{\glssymbolaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@symbolaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{symbol}{#1}{\glsentrysymbolaccess{#2}}}%
}
\newrobustcmd*{\glssymbolpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@symbolpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{symbolplural}{#1}{\glsentrysymbolpluralaccess{#2}}}%
}
\newrobustcmd*{\glsdescriptionaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@descaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{desc}{#1}{\glsentrydescaccess{#2}}}%
}
\newrobustcmd*{\glsdescriptionpluralaccessdisplay}[2]{%
```

```
  \ifcsundef{glo@\glsdetoklabel{#2}@descpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{descplural}{#1}{\glsentrydescpluralaccess{#2}}}%
}
\newrobustcmd*{\glsshortaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@shortaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{short}{#1}{\glsentryshortaccess{#2}}}%
}
\newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@shortpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glsentryshortpluralaccess{#2}}}%
}
\newrobustcmd*{\glslongaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@longaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{long}{#1}{\glsentrylongaccess{#2}}}%
}
\newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@longpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{longpl}{#1}{\glsentrylongpluralaccess{#2}}}%
}
\newrobustcmd*{\glsuseriaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@useriaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useri}{#1}{\glsentryuseriaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@useriiaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{userii}{#1}{\glsentryuseriiaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiiaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@useriiiaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useriii}{#1}{\glsentryuseriiiaccess{#2}}}%
}
\newrobustcmd*{\glsuserivaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#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}}%
            {\glssymbolpluralaccessdisplay
               {\glsentrysymbolplural{\glslabel}}{\glslabel}}
            {\glsinsert}%
        }%
        {%
          #1{\glsfirstpluralaccessdisplay
               {\glsentryfirstplural{\glslabel}}{\glslabel}}%
            {\glsdescriptionpluralaccessdisplay
               {\glsentrydescplural{\glslabel}}{\glslabel}}%
            {\glssymbolpluralaccessdisplay
               {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
            {\glsinsert}%
        }%
      }%
      {%
        \ifglsused\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}%
    }%
  }%
  {%
    \ifglsused\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
  {%
    \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
  {%
    #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
          {\glsentrytext{\glslabel}}{\glslabel}}%
        {\glsdescriptionaccessdisplay
          {\glsentrydesc{\glslabel}}{\glslabel}}%
        {\glssymbolaccessdisplay
          {\glsentrysymbol{\glslabel}}{\glslabel}}%
        {\glsinsert}}%
  }%
  {%
    \MakeUppercase{%
      #1{\glsfirstaccessdisplay
          {\glsentryfirst{\glslabel}}{\glslabel}}%
        {\glsdescriptionaccessdisplay
          {\glsentrydesc{\glslabel}}{\glslabel}}%
        {\glssymbolaccessdisplay
          {\glsentrysymbol{\glslabel}}{\glslabel}}%
```

```
            {\glsinsert}}%
        }%
      }%
    }%
  }%
  {%
    \ifglsused{\glslabel}%
    {%
      #2{\glscustomtext}%
        {\glsdescriptionaccessdisplay
          {\glsentrydesc{\glslabel}}{\glslabel}}%
        {\glssymbolaccessdisplay
          {\glsentrysymbol{\glslabel}}{\glslabel}}%
        {\glsinsert}%
    }%
    {%
      #1{\glscustomtext}%
        {\glsdescriptionaccessdisplay
          {\glsentrydesc{\glslabel}}{\glslabel}}%
        {\glssymbolaccessdisplay
          {\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}%
    }%
    {%
      \glsfirstpluralacessdisplay
         {\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}%
        }%
      }%
    }%
  }%
  {%
    \glscustomtext\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\glscapscase\@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@gls@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{\glsentrytext{#2}}{#2}#3}%
}
\def\@Glstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\Glsentrytext{#2}}{#2}#3}%
}
\def\@GLStext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glstextaccessdisplay{\mfirstucMakeUppercase{\glsentrytext{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\glsentryfirst{#2}}{#2}#3}%
}
\def\@Glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\Glsentryfirst{#2}}{#2}#3}%
}
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsfirstaccessdisplay{\mfirstucMakeUppercase{\glsentryfirst{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\glsentryplural{#2}}{#2}#3}%
}
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\Glsentryplural{#2}}{#2}#3}%
}
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glspluralaccessdisplay{\mfirstucMakeUppercase{\glsentryplural{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\glsentryfirstplural{#2}}{#2}#3}%
}
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\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\@glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
}
\def\@Glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}
\def\@GLSsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
    {\glssymbolaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbol{#2}}}{#2}%
     \mfirstucMakeUppercase{#3}}%
}
\def\@glssymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\glsentrysymbolplural{#2}}{#2}#3}%
}
\def\@Glssymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\Glsentrysymbolplural{#2}}{#2}#3}%
}
\def\@GLSsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
    {\glssymbolpluralaccessdisplay{\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\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
    {\glsuservaccessdisplay{\mfirstucMakeUppercase{\glsentryuserv{#2}}}{#2}%
     \mfirstucMakeUppercase{#3}}%
}
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\glsentryuservi{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\Glsentryuservi{#2}}{#2}#3}%
}
\def\@GLSuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
    {\glsuserviaccessdisplay{\mfirstucMakeUppercase{\glsentryuservi{#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{\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
        \acrnameformat{\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*{\showglonameaccess}[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@\glsdetoklabel{#1}@firstaccess\endcsname
}
\newcommand*{\showglofirstpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpluralaccess\endcsname
}
\newcommand*{\showglosymbolaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolaccess\endcsname
}
\newcommand*{\showglosymbolpluralaccess}[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}%
      }%
      {%
         \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}%
      }%
    }%
  }%
  {}%
```

8.4 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)]
\ifglsxindy
  \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{glshypernumber}
  \GlsAddXdyAttribute{hyperrm}
  \GlsAddXdyAttribute{hypersf}
  \GlsAddXdyAttribute{hypertt}
  \GlsAddXdyAttribute{hyperbf}
  \GlsAddXdyAttribute{hypermd}
  \GlsAddXdyAttribute{hyperit}
  \GlsAddXdyAttribute{hyperup}
  \GlsAddXdyAttribute{hypersl}
  \GlsAddXdyAttribute{hypersc}
  \GlsAddXdyAttribute{hyperemph}
\ifglsxindy
   \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]{%
\ifglsxindy
  \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{\theglsentrycounter}%
  \@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}{\theglsentrycounter}}%
\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%
}
\ifglsxindy
  \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\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"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
    \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
      \ifstrempty{##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[\glsentryitem{##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]{%
      \glsentryitem{##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[\glsentryitem{##1}\glstarget{##1}{##2}]}%
}%
\compatglossarystyle{long}{%
  \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{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{@glscompstyle@longragged}%
}%
\compatglossarystyle{longraggedheader}{%
 \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longraggedheaderborder}{%
 \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longragged3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glsentryitem{##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]{%
    \glsentryitem{##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\glsentryitem{##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{@glscompstyle@index}%
}%
\compatglossarystyle{indexhypergroup}{%
 \csuse{@glscompstyle@index}%
}%
\compatglossarystyle{tree}{%
  \renewcommand{\glossaryentryfield}[5]{%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glsentryitem{##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
    \glsentryitem{##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{alttree}{%
  \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]{%
       \glsentryitem{##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{alttreegroup}{%
 \csuse{@glscompstyle@alttree}%
}%
\compatglossarystyle{alttreehypergroup}{%
 \csuse{@glscompstyle@alttree}%
}%
\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{mcolalttree}{%
 \csuse{@glscompstyle@alttree}%
}%
\compatglossarystyle{mcolalttreegroup}{%
 \csuse{@glscompstyle@alttree}%
}%
\compatglossarystyle{mcolalttreehypergroup}{%
 \csuse{@glscompstyle@alttree}%
}%

\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]{%
    \glsentryitem{##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]{%
    \glsentryitem{##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]{%
  \ifcsempty{glo@\glsdetoklabel{#1}@prefix}%
  {#3}%
  {#2}%
}
\newcommand*{\ifglshasprefixplural}[3]{%
  \ifcsempty{glo@\glsdetoklabel{#1}@prefixplural}%
  {#3}%
  {#2}%
}
\newcommand*{\ifglshasprefixfirst}[3]{%
  \ifcsempty{glo@\glsdetoklabel{#1}@prefixfirst}%
  {#3}%
  {#2}%
}
\newcommand*{\ifglshasprefixfirstplural}[3]{%
  \ifcsempty{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}%
    {%
      \ifglshasprefix{#2}{\glsentryprefix{#2}\glsprefixsep}{}%
    }%
    {%
      \ifglshasprefixfirst{#2}{\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}{\glsentryprefixplural{#2}\glsprefixsep}{}%
    }%
    {%
      \ifglshasprefixfirstplural{#2}%
        {\glsentryprefixfirstplural{#2}\glsprefixsep}{}%
    }%
    \@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}%
      {%
        \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}%
    {%
      \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}}%
  \@glstarget{\glsnavhyperlinkname{#1}{#2}}{#3}%
  \expandafter\let
     \expandafter\@gls@list\csname @gls@hypergrouplist@#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@hypergrouprerun{%
      \GlossariesWarningNoLine{Navigation panel
      has changed. Rerun LaTeX}}%
  \fi
}
\let\gls@hypergrouprerun\relax
\AtEndDocument{\gls@hypergrouprerun}
\newcommand*{\@gls@hypergroup}[2]{%
\@ifundefined{@gls@hypergrouplist@#1}{%
   \expandafter\xdef\csname @gls@hypergrouplist@#1\endcsname{#2}%
}{%
   \expandafter\let\expandafter\@gls@tmp
      \csname @gls@hypergrouplist@#1\endcsname
   \expandafter\xdef\csname @gls@hypergrouplist@#1\endcsname{%
      \@gls@tmp,#2}%
}%
}
\newcommand*{\glsnavigation}{%
  \def\@gls@between{}%
  \ifcsundef{@gls@hypergrouplist@\@glo@type}%
```

```
  {%
    \def\@gls@list{}%
  }%
  {%
    \expandafter\let\expandafter\@gls@list
      \csname @gls@hypergrouplist@\@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*{\glssymbolnav}{%
 \glsnavhyperlink{glssymbols}{\glsgetgrouptitle{glssymbols}}%
 \glshypernavsep
 \glsnavhyperlink{glsnumbers}{\glsgetgrouptitle{glsnumbers}}%
 \glshypernavsep
}
```

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{}%
    }%
    {\glspostinline}%
 \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[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}
```

8.10 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][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}}]}%
}
```

8.11 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\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}{|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]{%
    \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}{\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}{llll}}%
    {\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
  }%
  \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}{|l|p{\glsdescwidth}|l|p{\glspagelistwidth}|}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheaderborder}{%
  \setglossarystyle{long4colheaderborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|p{\glsdescwidth}|l|p{\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}%
  \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}%
  \ifglsnogroupskip
    \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}%
  \ifglsnogroupskip
    \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\vsize\@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\vsize\@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}{l>{\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}{l>{\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
  }%
  \ifglsnogroupskip
    \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}%
}
```

8.14 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*{\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{mcolalttree}{%
  \setglossarystyle{alttree}%
  \renewenvironment{theglossary}%
  {%
     \begin{multicols}{\glsmcols}
     \def\@gls@prevlevel{-1}%
     \mbox{}\par
  }%
  {\par\end{multicols}}%
}
\newglossarystyle{mcolalttreegroup}{%
  \setglossarystyle{mcolalttree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcolalttreehypergroup}{%
  \setglossarystyle{mcolalttree}%
  \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{mcolalttreespannav}{%
  \setglossarystyle{alttree}%
  \renewenvironment{theglossary}%
  {%
     \begin{multicols}{\glsmcols}[\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]{%
    \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{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}{lp{\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}{llll}}%
    {\end{supertabular}}%
}
\newglossarystyle{super4colborder}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
     \begin{supertabular}{|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|}}%
    {\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}%
       {|l|p{\glsdescwidth}|l|p{\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}%
      {|l|p{\glsdescwidth}|l|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}{l>{\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}|}}%
    {\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}}%
}
```

8.17 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\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}{\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}%
      {}%
      {%
```

```
        \settowidth{\dimen@}%
         {\glstreenamefmt{\glsentryname{\@glo@label}}}%
        \ifdim\dimen@>\gls@tmplen
          \gls@tmplen=\dimen@
          \letcs{\@glswidestname}{glo@\glsdetoklabel{\@glo@label}@name}%
        \fi
      }%
    }%
  }%
}
\newcommand*{\glssetwidest}[2][0]{%
  \expandafter\def\csname @glswidestname\romannumeral#1\endcsname{%
    #2}%
}
\newcommand*{\@glswidestname}{}
\newcommand*{\glstreenamebox}[2]{%
  \makebox[#1][l]{#2}%
}
\newglossarystyle{alttree}{%
  \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[0pt][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[0pt][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{alttreegroup}{%
  \setglossarystyle{alttree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
    \par\indexspace}%
}
\newglossarystyle{alttreehypergroup}{%
  \setglossarystyle{alttree}%
  \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\@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]}%
   \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]{%
  \ifstrempty{#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]{}
\glssavenumberlistfalse
\newcommand*\@glo@seeautonumberlist{}
\@gls@declareoption{seeautonumberlist}{%
   \renewcommand*{\@glo@seeautonumberlist}{%
      \def\@glo@prefix{\glsnextpages}%
   }%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{}
\glsesclocationstrue
\newcommand*{\@gls@loadlong}{\RequirePackage{glossary-long}[=v4.49]}
\@gls@declareoption{nolong}{\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{nolist}{%
 \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]{}
\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
      \ifglsentrycounter
        \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*\glssortnumberfmt[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*{\@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]{%
  \protected@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@glsisacronymlist
\newcommand*{\gls@checkisacronymlist}[1]{%
   \glsIfListOfAcronyms{#1}%
     {\@glsisacronymlisttrue}{\@glsisacronymlistfalse}%
}
\newcommand*{\SetAcronymLists}[1]{%
  \renewcommand*{\@glsacronymlists}{#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 `theglossary' 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}{%
  \ifglssanitizesort
    \@@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}%
    \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
    }%
    {}%
  }
}
\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\@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}
\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}[=v4.49]
\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*{\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}
  \@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}%
  \ifglsxindy
    \def\istfilename{\gls@istfilebase.xdy}
  \else
    \def\istfilename{\gls@istfilebase.ist}
  \fi
}
\@onlypremakeg\setStyleFile
\ifglsxindy
  \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]{%
    \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]{\glshypernumber{#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}%
  }%
  \glsglossarymark{\glossarytoctitle}%
}
\ifcsundef{glossarymark}%
{%
  \newcommand{\glsglossarymark}[1]{\glossarymark{#1}}
}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand{\glsglossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUChead{#1}}{\memUChead{#1}}%
      \else
        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand{\glsglossarymark}[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@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]{%
    \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@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{}
\ifglsxindy
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
  \edef\@xdyuseralphabets{%
    \@xdyuseralphabets ^^J
    (define-alphabet "#1" (#2))}}
\else
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
     \glsnoxindywarning\GlsAddXdyAlphabet}
\fi
\ifglsxindy
  \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]{%
     \glsnoxindywarning\GlsAddXdyLocation}
\fi
\ifglsxindy
  \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
\ifglsxindy
  \newcommand*\GlsSetXdyLocationClassOrder[1]{%
    \def\@xdylocationclassorder{#1}}
\else
  \newcommand*\GlsSetXdyLocationClassOrder[1]{%
    \glsnoxindywarning\GlsSetXdyLocationClassOrder}
```

```
\fi
\ifglsxindy
  \def\@xdysortrules{}
\fi
\ifglsxindy
  \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]{%
    \glsnoxindywarning\GlsAddSortRule}
\fi
\ifglsxindy
  \def\@xdyrequiredstyles{tex}
\fi
\ifglsxindy
  \newcommand*\GlsAddXdyStyle[1]{%
    \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
  \newcommand*\GlsAddXdyStyle[1]{%
    \glsnoxindywarning\GlsAddXdyStyle}
\fi
\ifglsxindy
  \newcommand*\GlsSetXdyStyles[1]{%
    \edef\@xdyrequiredstyles{#1}}
\else
  \newcommand*\GlsSetXdyStyles[1]{%
    \glsnoxindywarning\GlsSetXdyStyles}
\fi
\newcommand*{\findrootlanguage}{}
\def\@xdylanguage#1#2{}
\ifglsxindy
  \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{}
\ifglsxindy
  \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*{\forglsentries}[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@}%
  {%
    \forglsentries[\@@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\forglsentries\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}%
  {%
    \ifcsempty{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
  }%
}
\newcommand*{\ifglshasdesc}[3]{%
  \ifcsempty{glo@\glsdetoklabel{#1}@desc}%
  {#3}%
  {#2}%
}
\newcommand*{\ifglsdescsuppressed}[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@\glsdetoklabel{#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@\glsdetoklabel{#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*{\defglsentryfmt}[2][\glsdefaulttype]{%
  \csgdef{gls@#1@entryfmt}{#2}%
}
\newcommand*{\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*{\@gls@forbidtexext}[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@forbidtexext\csname @glotype@#2@log\endcsname
  \expandafter\@gls@forbidtexext\csname @glotype@#2@in\endcsname
  \expandafter\@gls@forbidtexext\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@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*{\@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]{%
  ??%
  \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*{\glsexpandfields}{%
  \let\gls@assign@field\@gls@expand@fields
}
\newcommand*{\glsnoexpandfields}{%
  \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@restoreat{\noexpand\catcode`\noexpand\@=\number\catcode`\@\relax}%
  \makeatletter
  \InputIfFileExists{\jobname.glsdefs}{}{}%
  \@gls@restoreat
  \undef\@gls@restoreat
  \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
      }%
    }%
    {}%
  }%
  \glswritedefhook
  \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{\glsfieldedef}[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{\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}{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{\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}{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}%
  {%
     \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
  \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\@gloname\@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\@glodesc\@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{,}%
  \forglsentries[\glo@type]{\glo@label}%
   {%
     \ifdefequal\@glo@thislabel\glo@label
       {}{\protected@eappto\glo@list{\glo@label,}}%
   }%
  \cslet{glolist@\glo@type}{\glo@list}%
  \csdef{glo@\@glo@thislabel @type}{#2}%
}
\ifglsxindy
  \newcommand*{\@glossaryentryfield}{\string\\glossentry}
\else
  \newcommand*{\@glossaryentryfield}{\string\glossentry}
\fi
\ifglsxindy
  \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}%
  \ifglsxindy
    \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\@cgls@##1##2[##3]{%
   \ifnum\glsentryprevcount{##2}=1\relax
```

```
    \cglsformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@gls@{##1}{##2}[##3]%
  \fi
}%
\def\@cglspl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cglsplformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@glspl@{##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\@cGlspl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cGlsplformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@Glspl@{##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}%
 \@gls@{#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}%
 \@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}
\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}{%
  \@@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
      {%
        \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
        {%
```

```
          \acronymfont{\glsentryshort{\glslabel}}\glsinsert
        }%
        {%
          \acronymfont{\Glsentryshort{\glslabel}}\glsinsert
        }%
        {%
          \mfirstucMakeUppercase
            {\acronymfont{\glsentryshort{\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
  }%
}
\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}}%
      }%
      {%
        \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
      \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@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@dosubst{%
      \noexpand\DTLsubstituteall\noexpand\gls@xdystring
      {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
    }%
```

```
    \gls@dosubst
  }%
  \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}%
    \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@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
}
\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]{%
  \@glsshowtarget{#1}%
  \settoheight{\gls@tmplen}{#2}%
  \raisebox{\gls@tmplen}{\hypertarget{#1}{}}#2%
}
\newcommand*{\glsdohyperlink}[2]{%
 \@glsshowtarget{#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\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*{\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\glscapscase\@secondofthree
    \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@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \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\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\glscapscase\@secondofthree
    \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\glscapscase\@thirdofthree
    \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*{\glsdisp}{\@gls@hyp@opt\@glsdisp}
\newcommand*{\@glsdisp}[3][]{%
  \glsdoifexists{#2}{%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@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*{\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*{\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*{\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*{\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*{\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*{\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*{\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\glscapscase\@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\glscapscase\@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{\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\@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{\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}
\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\@secondofthree
    \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}
\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\glscapscase\@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@acrentryname}[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
  }%
  {%
    \@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*{\glshyperlink}[2][\glsentrytext{\@glo@label}]{%
 \def\@glo@label{#2}%
 \@glslink{\glolinkprefix\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\glstildechar{\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]{%
    \glsnoxindywarning\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]{%
    \glsnoxindywarning\GlsSetXdyNumberGroupOrder}
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
    \@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*{\@glsminrange}{2}
\ifglsxindy
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \renewcommand*{\@glsminrange}{#1}}
\else
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
```

```
    \glsnoxindywarning\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\glsreseteentrylist
        \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
   \\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"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
    \@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#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@checkescbar####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
}
\ifglsxindy
 \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{@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
  \let\warn@noprintglossary\@makeglossaries@warn@noprintglossary
  \ifglssavenumberlist
    \edef\@gls@dodeflistparser{\noexpand\DeclareListParser
      {\noexpand\glsnumlistparser}{\delimN}}%
    \@gls@dodeflistparser
  \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*{\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
}%
}
\@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*{\@glswritefiles}{%
  \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\begingroup\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]{%
  \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}{%
  \@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]{%
  \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]{%
  \ifglsesclocations
    \@@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{\theHglsentrycounter}%
  \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{\theglsentrycounter}%
  \endgroup
  \@gls@checkmkidxchars\@glslocref
  \expandafter\ifx\theHglsentrycounter\theglsentrycounter\relax
    \def\@glo@counterprefix{}%
  \else
    \protected@edef\@glsHlocref{\theHglsentrycounter}%
    \@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}{%
  \ifglsxindy
    \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
    \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^^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
      :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@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}{%
  \forallglossaries{\@@glo@type}{\printnoidxglossary[type=\@@glo@type]}%
}
\newcommand*{\@printgloss@setsort}{}
\newcommand*{\@gls@preglossaryhook}{}
\newcommand{\@printglossary}[2]{%
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{\csname @glotype@\@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@dotoctitle{\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 @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@dotoctitle
    \@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
    \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{}%
  \forglsentries[#1]{\@gls@thislabel}%
  {%
    \xifinlistcs{\@gls@thislabel}{@glsref@#1}%
    {%
      \listeadd{\@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@\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@dotoctitle\relax
}
\define@key{printgloss}{toctitle}{%
 \def\glossarytoctitle{#1}%
 \let\gls@dotoctitle\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{glssubentrycounter#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*{\glsreseteentrycounter}{%
  \ifglsentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*{\glsstepentry}[1]{%
  \ifglsentrycounter
    \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]{%
 \ifglsentrycounter
   \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
 \else
   \ifglssubentrycounter
     \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
   \else
     \gls{#1}%
   \fi
 \fi
}
\newcommand*{\glsentrycounterlabel}{%
 \ifglsentrycounter
   \theglossaryentry.\space
 \fi
}
\newcommand*{\glssubentrycounterlabel}{%
 \ifglssubentrycounter
   \theglossarysubentry)\space
 \fi
}
```

```
\newcommand*{\glsentryitem}[1]{%
  \ifglsentrycounter
    \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*{\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]{%
  \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}%
  {%
     \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}{glssymbols}}
            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}{\glssymbolsgroupname}}{glssymbols}{%
\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\csuse{@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\glshypernumber#1{#1}%
}%
{%
  \def\glshypernumber#1{\@glshypernumber#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\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\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}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \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}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \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}{\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]{%
    \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
        }%
        {%
          \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}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
```

```
\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@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
  \@@gls@noexpand@assign@default@plurals
}
\newcommand*{\@@gls@noexpand@assign@default@plurals}{%
  \expandafter\ifx\csname glo@\the\glslabeltok @plural\endcsname\@@gls@@default@acr@plural
    \csxdef{glo@\the\glslabeltok @plural}{\expandonce\@glo@shortpl}%
  \fi
  \expandafter\ifx\csname glo@\the\glslabeltok @firstpl\endcsname\@@gls@@default@acr@firstplural
    \csxdef{glo@\the\glslabeltok @firstpl}{%
      \expandonce\@glo@longpl\glsspace(\expandonce\@glo@shortpl)}%
  \fi
  \expandafter\ifx\csname glo@\the\glslabeltok @descplural\endcsname\@@gls@@default@acr@descplural
    \csxdef{glo@\the\glslabeltok @descplural}{\expandonce\@glo@longpl}%
  \fi
}
\newcommand*{\@@gls@@default@acr@plural}{\expandonce\@glo@shortpl}
\newcommand*{\@@gls@@default@acr@firstplural}{%
 \expandonce\@glo@longpl\glsspace(\expandonce\@glo@shortpl)%
}
\newcommand*{\@@gls@@default@acr@descplural}{\expandonce\@glo@longpl}
\newcommand*{\SetDefaultAcronymStyle}{%
  \@for\@gls@type:=\@glsacronymlists\do{%
    \SetDefaultAcronymDisplayStyle{\@gls@type}%
  }%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@glsacronymlists\@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}{\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}%
        \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}{%
  \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
            {\glscapscase
             {\@glo@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}{%
  \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
        \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}{%
  \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\glscustomtext
    {%
      \let\gls@org@insert\glsinsert
      \let\glsinsert\@empty
      \ifglsused{\glslabel}%
      {%
        \acronymfont{\glsgenentryfmt}\gls@org@insert
      }%
      {%
        \glsgenentryfmt
        \ifglshassymbol{\glslabel}%
        {%
          \glsifplural
          {%
            \def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%
          }%
          {%
            \def\@glo@symbol{\glsentrysymbol{\glslabel}}%
          }%
          \space
            (\glscapscase
            {\firstacronymfont{\@glo@symbol}}%
            {\firstacronymfont{\@glo@symbol}}%
            {\firstacronymfont{\mfirstucMakeUppercase{\@glo@symbol}}})%
        }%
        {}%
      }%
    }%
    {\glscustomtext\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}{%
  \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@gls@assign@firstpl\gls@assign@firstpl
  \let\@org@gls@assign@plural\gls@assign@plural
  \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
  \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@symbolplural##1##2{%
    \@@gls@expand@field{##1}{symbolplural}{##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
  \let\gls@assign@descplural\@org@gls@assign@descplural
}
\newcommand*{\SetDUAStyle}{%
  \@gls@deprecated@acr@warning{dua}{dua}{long-noshort}%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@glsacronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetDUADisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \DUANewAcronymDef
  }%
  \@for\@gls@type:=\@glsacronymlists\do{%
    \SetDUADisplayStyle{\@gls@type}%
  }%
}
\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}%
}
\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
          \SetDUAStyle
        \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\@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}[=v4.49]
\@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@\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*{\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@\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 @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 glolist@#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 @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}%
}
\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*{\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*{\glstextaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@textaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{text}{#1}{\glsentrytextaccess{#2}}}%
}
\newrobustcmd*{\glspluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@pluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{plural}{#1}{\glsentrypluralaccess{#2}}}%
}
\newrobustcmd*{\glsfirstaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@firstaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{first}{#1}{\glsentryfirstaccess{#2}}}%
}
\newrobustcmd*{\glsfirstpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@firstpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{firstpl}{#1}{\glsentryfirstpluralaccess{#2}}}%
}
\newrobustcmd*{\glssymbolaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@symbolaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{symbol}{#1}{\glsentrysymbolaccess{#2}}}%
}
\newrobustcmd*{\glssymbolpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@symbolpluralaccess}%
```

```
  {#1}%
  {\@gls@fieldaccess@display{#2}{symbolplural}{#1}{\glsentrysymbolpluralaccess{#2}}}%
}
\newrobustcmd*{\glsdescriptionaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@descaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{desc}{#1}{\glsentrydescaccess{#2}}}%
}
\newrobustcmd*{\glsdescriptionpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@descpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{descplural}{#1}{\glsentrydescpluralaccess{#2}}}%
}
\newrobustcmd*{\glsshortaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@shortaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{short}{#1}{\glsentryshortaccess{#2}}}%
}
\newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@shortpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glsentryshortpluralaccess{#2}}}%
}
\newrobustcmd*{\glslongaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@longaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{long}{#1}{\glsentrylongaccess{#2}}}%
}
\newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@longpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{longpl}{#1}{\glsentrylongpluralaccess{#2}}}%
}
\newrobustcmd*{\glsuseriaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@useriaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useri}{#1}{\glsentryuseriaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@useriiaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{userii}{#1}{\glsentryuseriiaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiiaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@useriiiaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useriii}{#1}{\glsentryuseriiiaccess{#2}}}%
}
\newrobustcmd*{\glsuserivaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#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}{userv i}{#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}}%
            {\glssymbolpluralaccessdisplay
               {\glsentrysymbolplural{\glslabel}}{\glslabel}}
            {\glsinsert}%
        }%
        {%
          #1{\glsfirstpluralaccessdisplay
               {\glsentryfirstplural{\glslabel}}{\glslabel}}%
            {\glsdescriptionpluralaccessdisplay
               {\glsentrydescplural{\glslabel}}{\glslabel}}%
            {\glssymbolpluralaccessdisplay
               {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
            {\glsinsert}%
        }%
      }%
```

```
  {%
    \ifglsused\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}%
    }%
  }%
  {%
    \ifglsused\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
  {%
    \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
  {%
    #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
           {\glsentrytext{\glslabel}}{\glslabel}}%
         {\glsdescriptionaccessdisplay
           {\glsentrydesc{\glslabel}}{\glslabel}}%
         {\glssymbolaccessdisplay
           {\glsentrysymbol{\glslabel}}{\glslabel}}%
         {\glsinsert}}%
```

```
        }%
        {%
          \MakeUppercase{%
            #1{\glsfirstaccessdisplay
                {\glsentryfirst{\glslabel}}{\glslabel}}%
              {\glsdescriptionaccessdisplay
                {\glsentrydesc{\glslabel}}{\glslabel}}%
              {\glssymbolaccessdisplay
                {\glsentrysymbol{\glslabel}}{\glslabel}}%
              {\glsinsert}}%
        }%
      }%
    }%
  }%
  {%
    \ifglsused{\glslabel}%
    {%
      #2{\glscustomtext}%
        {\glsdescriptionaccessdisplay
          {\glsentrydesc{\glslabel}}{\glslabel}}%
        {\glssymbolaccessdisplay
          {\glsentrysymbol{\glslabel}}{\glslabel}}%
        {\glsinsert}%
    }%
    {%
      #1{\glscustomtext}%
        {\glsdescriptionaccessdisplay
          {\glsentrydesc{\glslabel}}{\glslabel}}%
        {\glssymbolaccessdisplay
          {\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}%
    }%
    {%
      \glsfirstpluralacessdisplay
         {\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}%
        }%
      }%
    }%
  }%
  {%
    \glscustomtext\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\glscapscase\@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@gls@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{\glsentrytext{#2}}{#2}#3}%
}
\def\@Glstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\Glsentrytext{#2}}{#2}#3}%
}
\def\@GLStext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glstextaccessdisplay{\mfirstucMakeUppercase{\glsentrytext{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\glsentryfirst{#2}}{#2}#3}%
}
\def\@Glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\Glsentryfirst{#2}}{#2}#3}%
}
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsfirstaccessdisplay{\mfirstucMakeUppercase{\glsentryfirst{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\glsentryplural{#2}}{#2}#3}%
}
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\Glsentryplural{#2}}{#2}#3}%
}
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glspluralaccessdisplay{\mfirstucMakeUppercase{\glsentryplural{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\glsentryfirstplural{#2}}{#2}#3}%
}
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\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\@glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
}
\def\@Glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}
\def\@GLSsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
    {\glssymbolaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbol{#2}}}{#2}%
     \mfirstucMakeUppercase{#3}}%
}
\def\@glssymbolplural@#1#2[#3]{%
```

```
  \@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\glsentrysymbolplural{#2}}{#2}#3}%
}
\def\@Glssymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\Glsentrysymbolplural{#2}}{#2}#3}%
}
\def\@GLSsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
    {\glssymbolpluralaccessdisplay{\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\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
    {\glsuservaccessdisplay{\mfirstucMakeUppercase{\glsentryuserv{#2}}}{#2}%
     \mfirstucMakeUppercase{#3}}%
}
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\glsentryuservi{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\Glsentryuservi{#2}}{#2}#3}%
}
\def\@GLSuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
    {\glsuserviaccessdisplay{\mfirstucMakeUppercase{\glsentryuservi{#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{\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}{%
  \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@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
  \@@gls@noexpand@assign@default@plurals
}
\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
        \acrnameformat{\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*{\showglonameaccess}[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@\glsdetoklabel{#1}@firstaccess\endcsname
}
\newcommand*{\showglofirstpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpluralaccess\endcsname
}
\newcommand*{\showglosymbolaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolaccess\endcsname
}
\newcommand*{\showglosymbolpluralaccess}[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
}
```

8.20 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
```

```
        Language modules need to be installed separately.\MessageBreak
        Please check on CTAN for a bundle called\MessageBreak
       'glossaries-\CurrentTrackedLanguage' or similar}%
    }%
  }%
}%
{}%
```

8.21 Rollback v4.49 (glossaries-compatible-207-2021-11-01.sty)

Version 4.49 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-207}[2021/11/01 v4.49 (NLCT)]
\ifglsxindy
  \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{glshypernumber}
  \GlsAddXdyAttribute{hyperrm}
  \GlsAddXdyAttribute{hypersf}
  \GlsAddXdyAttribute{hypertt}
  \GlsAddXdyAttribute{hyperbf}
  \GlsAddXdyAttribute{hypermd}
  \GlsAddXdyAttribute{hyperit}
```

```
  \GlsAddXdyAttribute{hyperup}
  \GlsAddXdyAttribute{hypersl}
  \GlsAddXdyAttribute{hypersc}
  \GlsAddXdyAttribute{hyperemph}
\ifglsxindy
   \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]{%
\ifglsxindy
  \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{\theglsentrycounter}%
  \@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}{\theglsentrycounter}}%
\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%
}
```

```
\ifglsxindy
  \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\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\glsreseteentrylist
     \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"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
    \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
      \ifstrempty{##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[\glsentryitem{##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]{%
      \glsentryitem{##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[\glsentryitem{##1}\glstarget{##1}{##2}]}%
}%
\compatglossarystyle{long}{%
  \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{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{@glscompstyle@longragged}%
}%
\compatglossarystyle{longraggedheader}{%
 \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longraggedheaderborder}{%
 \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longragged3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glsentryitem{##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]{%
    \glsentryitem{##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\glsentryitem{##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{@glscompstyle@index}%
}%
\compatglossarystyle{indexhypergroup}{%
 \csuse{@glscompstyle@index}%
}%
\compatglossarystyle{tree}{%
  \renewcommand{\glossaryentryfield}[5]{%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glsentryitem{##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
    \glsentryitem{##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{alttree}{%
  \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]{%
     \glsentryitem{##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{alttreegroup}{%
```

```
 \csuse{@glscompstyle@alttree}%
}%
\compatglossarystyle{alttreehypergroup}{%
 \csuse{@glscompstyle@alttree}%
}%
\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{mcolalttree}{%
 \csuse{@glscompstyle@alttree}%
}%
\compatglossarystyle{mcolalttreegroup}{%
 \csuse{@glscompstyle@alttree}%
}%
\compatglossarystyle{mcolalttreehypergroup}{%
 \csuse{@glscompstyle@alttree}%
}%

\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]{%
    \glsentryitem{##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]{%
    \glsentryitem{##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]{%
  \ifcsempty{glo@\glsdetoklabel{#1}@prefix}%
  {#3}%
  {#2}%
}
\newcommand*{\ifglshasprefixplural}[3]{%
  \ifcsempty{glo@\glsdetoklabel{#1}@prefixplural}%
  {#3}%
```

```
  {#2}%
}
\newcommand*{\ifglshasprefixfirst}[3]{%
  \ifcsempty{glo@\glsdetoklabel{#1}@prefixfirst}%
  {#3}%
  {#2}%
}
\newcommand*{\ifglshasprefixfirstplural}[3]{%
  \ifcsempty{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}%
    {%
      \ifglshasprefix{#2}{\glsentryprefix{#2}\glsprefixsep}{}%
    }%
    {%
      \ifglshasprefixfirst{#2}{\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}{\glsentryprefixplural{#2}\glsprefixsep}{}%
    }%
    {%
      \ifglshasprefixfirstplural{#2}%
        {\glsentryprefixfirstplural{#2}\glsprefixsep}{}%
    }%
    \@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}%
      {%
        \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}%
    {%
      \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]%
  }%
}
```

8.25 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}}%
  \@glstarget{\glsnavhyperlinkname{#1}{#2}}{#3}%
  \expandafter\let
     \expandafter\@gls@list\csname @gls@hypergrouplist@#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@hypergrouprerun{%
      \GlossariesWarningNoLine{Navigation panel
      has changed. Rerun LaTeX}}%
  \fi
}
\let\gls@hypergrouprerun\relax
\AtEndDocument{\gls@hypergrouprerun}
\newcommand*{\@gls@hypergroup}[2]{%
\@ifundefined{@gls@hypergrouplist@#1}{%
   \expandafter\xdef\csname @gls@hypergrouplist@#1\endcsname{#2}%
}{%
```

```
   \expandafter\let\expandafter\@gls@tmp
      \csname @gls@hypergrouplist@#1\endcsname
   \expandafter\xdef\csname @gls@hypergrouplist@#1\endcsname{%
      \@gls@tmp,#2}%
}%
}
\newcommand*{\glsnavigation}{%
  \def\@gls@between{}%
  \ifcsundef{@gls@hypergrouplist@\@glo@type}%
  {%
    \def\@gls@list{}%
  }%
  {%
    \expandafter\let\expandafter\@gls@list
      \csname @gls@hypergrouplist@\@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*{\glssymbolnav}{%
 \glsnavhyperlink{glssymbols}{\glsgetgrouptitle{glssymbols}}%
 \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*{\glsgroupheading}[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}{|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]{%
    \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}{\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}{llll}}%
    {\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
  }%
  \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}{|l|p{\glsdescwidth}|l|p{\glspagelistwidth}|}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheaderborder}{%
  \setglossarystyle{long4colheaderborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|p{\glsdescwidth}|l|p{\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}%
  \ifglsnogroupskip
    \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}%
  \ifglsnogroupskip
    \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\vsize\@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\vsize\@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}{l>{\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}{l>{\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
  }%
  \ifglsnogroupskip
    \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{mcolalttree}{%
  \setglossarystyle{alttree}%
  \renewenvironment{theglossary}%
  {%
     \begin{multicols}{\glsmcols}
     \def\@gls@prevlevel{-1}%
     \mbox{}\par
  }%
  {\par\end{multicols}}%
}
\newglossarystyle{mcolalttreegroup}{%
  \setglossarystyle{mcolalttree}%
```

```
  \renewcommand{\glsgroupheading}[1]{\par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcolalttreehypergroup}{%
  \setglossarystyle{mcolalttree}%
  \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{mcolalttreespannav}{%
  \setglossarystyle{alttree}%
  \renewenvironment{theglossary}%
  {%
     \begin{multicols}{\glsmcols}[\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{\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{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}{lp{\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}{llll}}%
    {\end{supertabular}}%
}
\newglossarystyle{super4colborder}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
     \begin{supertabular}{|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|}}%
    {\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}%
       {|l|p{\glsdescwidth}|l|p{\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}%
       {|l|p{\glsdescwidth}|l|p{\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}{l>{\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}|}}%
    {\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}{\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}%
      {}%
      {%
        \settowidth{\dimen@}%
         {\glstreenamefmt{\glsentryname{\@glo@label}}}%
        \ifdim\dimen@>\gls@tmplen
          \gls@tmplen=\dimen@
          \letcs{\@glswidestname}{glo@\glsdetoklabel{\@glo@label}@name}%
        \fi
      }%
    }%
  }%
}
\newcommand*{\glssetwidest}[2][0]{%
  \expandafter\def\csname @glswidestname\romannumeral#1\endcsname{%
    #2}%
}
\newcommand*{\@glswidestname}{}
\newcommand*{\glstreenamebox}[2]{%
  \makebox[#1][l]{#2}%
}
\newglossarystyle{alttree}{%
```

```
\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[0pt][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[0pt][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{alttreegroup}{%
  \setglossarystyle{alttree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
    \par\indexspace}%
}
\newglossarystyle{alttreehypergroup}{%
  \setglossarystyle{alttree}%
  \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{\glssentencecase}[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\@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]{%
 \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]{%
  \ifstrempty{#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]{}
\glssavenumberlistfalse
\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{nolong}{\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{nolist}{%
 \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
      \ifglsentrycounter
        \newcounter{glossarysubentry}[glossaryentry]%
      \else
        \newcounter{glossarysubentry}%
      \fi
      \def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
    \fi
  }%
  {}%
}
\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*\glssortnumberfmt[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][]{%
      \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]{%
  \protected@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@glsisacronymlist
\newcommand*{\gls@checkisacronymlist}[1]{%
   \glsIfListOfAcronyms{#1}%
     {\@glsisacronymlisttrue}{\@glsisacronymlistfalse}%
}
\newcommand*{\SetAcronymLists}[1]{%
  \renewcommand*{\@glsacronymlists}{#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 `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}{%
  \ifglssanitizesort
    \@@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
      \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]{%
  \ifglssavewrites
    \renewcommand*{\glswritefiles}{\@glswritefiles}%
  \else
    \let\glswritefiles\@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}{\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*{\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
  \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]
  \ifdefvoid\@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}%
```

```
  \ifglsxindy
    \def\istfilename{\gls@istfilebase.xdy}
  \else
    \def\istfilename{\gls@istfilebase.ist}
  \fi
}
\@onlypremakeg\setStyleFile
\ifglsxindy
  \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]{%
    \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]{\glshypernumber{#1}}%
}
\newcommand{\glslocationcstoencap}[3]{\csuse{#2}{\csuse{#1}{#3}}}
\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}%
  }%
  \glsglossarymark{\glossarytoctitle}%
}
\ifcsundef{glossarymark}%
{%
  \newcommand{\glsglossarymark}[1]{\glossarymark{#1}}
}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand{\glsglossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUChead{#1}}{\memUChead{#1}}%
      \else
        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand{\glsglossarymark}[1]{%
      \ifglsucmark
        \@mkboth{\glsuppercase{#1}}{\glsuppercase{#1}}%
      \else
        \@mkboth{#1}{#1}%
      \fi
    }
  }
}
\providecommand{\glossarymark}[1]{%
  \ifglsucmark
    \@mkboth{\glsuppercase{#1}}{\glsuppercase{#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@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]{%
    \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"#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{}
\ifglsxindy
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
  \edef\@xdyuseralphabets{%
    \@xdyuseralphabets ^^J
    (define-alphabet "#1" (#2))}}
\else
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
     \glsnoxindywarning\GlsAddXdyAlphabet}
\fi
\ifglsxindy
  \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]{%
    \glsnoxindywarning\GlsAddXdyLocation}
\fi
\ifglsxindy
 \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
\ifglsxindy
 \newcommand*\GlsSetXdyLocationClassOrder[1]{%
   \def\@xdylocationclassorder{#1}}
\else
 \newcommand*\GlsSetXdyLocationClassOrder[1]{%
   \glsnoxindywarning\GlsSetXdyLocationClassOrder}
\fi
\ifglsxindy
 \def\@xdysortrules{}
\fi
\ifglsxindy
 \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]{%
   \glsnoxindywarning\GlsAddSortRule}
```

```
\fi
\ifglsxindy
  \def\@xdyrequiredstyles{tex}
\fi
\ifglsxindy
  \newcommand*\GlsAddXdyStyle[1]{%
    \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
  \newcommand*\GlsAddXdyStyle[1]{%
    \glsnoxindywarning\GlsAddXdyStyle}
\fi
\ifglsxindy
  \newcommand*\GlsSetXdyStyles[1]{%
    \edef\@xdyrequiredstyles{#1}}
\else
  \newcommand*\GlsSetXdyStyles[1]{%
    \glsnoxindywarning\GlsSetXdyStyles}
\fi
\def\@xdylanguage#1#2{}
\ifglsxindy
  \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{}
\ifglsxindy
  \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*{\forglsentries}[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@}%
  {%
    \forglsentries[\@@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\forglsentries\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}%
  {%
    \ifcsempty{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
  }%
}
\newcommand*{\ifglshasdesc}[3]{%
  \ifcsempty{glo@\glsdetoklabel{#1}@desc}%
  {#3}%
  {#2}%
}
\newcommand*{\ifglsdescsuppressed}[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@\glsdetoklabel{#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@\glsdetoklabel{#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@forbidtexext}[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@forbidtexext\csname @glotype@#2@log\endcsname
  \expandafter\@gls@forbidtexext\csname @glotype@#2@in\endcsname
  \expandafter\@gls@forbidtexext\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*{\@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]{%
  ??%
  \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*{\glsexpandfields}{%
  \let\gls@assign@field\@gls@expand@fields
}
\newcommand*{\glsnoexpandfields}{%
  \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@restoreat{\noexpand\catcode`\noexpand\@=\number\catcode`\@\relax}%
  \makeatletter
  \InputIfFileExists{\jobname.glsdefs}{}{}%
  \@gls@restoreat
  \undef\@gls@restoreat
  \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
      }%
    }%
    {}%
  }%
  \glswritedefhook
  \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{\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}{}%
   }%
 }%
}
\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
  \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\@gloname\@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\@glodesc\@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{,}%
  \forglsentries[\glo@type]{\glo@label}%
   {%
     \ifdefequal\@glo@thislabel\glo@label
       {}{\protected@eappto\glo@list{\glo@label,}}%
   }%
  \cslet{glolist@\glo@type}{\glo@list}%
  \csdef{glo@\@glo@thislabel @type}{#2}%
}
\ifglsxindy
  \newcommand*{\@glossaryentryfield}{\string\\glossentry}
\else
  \newcommand*{\@glossaryentryfield}{\string\glossentry}
\fi
\ifglsxindy
  \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}%
  \ifglsxindy
    \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*{\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\@cgls@##1##2[##3]{%
   \ifnum\glsentryprevcount{##2}=1\relax
```

```
    \cglsformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@gls@{##1}{##2}[##3]%
  \fi
}%
\def\@cglspl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cglsplformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@glspl@{##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\@cGlspl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cGlsplformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@Glspl@{##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}%
 \@gls@{#1}{#2}[#3]%
}
\newcommand*{\cglsformat}[2]{%
  \ifglshaslong{#1}{\glsentrylong{#1}}{\glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cGls}{\@gls@hyp@opt\@cGls}
\glsmfuaddmap{\cgls}{\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
```

```
      {%
        \ifglsused\glslabel
        {%
          \glsentrytext{\glslabel}\glsinsert
        }%
        {%
          \glsentryfirst{\glslabel}\glsinsert
        }%
      }%
      {%
        \ifglsused\glslabel
        {%
           \Glsentrytext{\glslabel}\glsinsert
        }%
        {%
          \Glsentryfirst{\glslabel}\glsinsert
        }%
      }%
      {%
        \ifglsused\glslabel
        {%
          \glsuppercase{\glsentrytext{\glslabel}\glsinsert}%
        }%
        {%
          \glsuppercase{\glsentryfirst{\glslabel}\glsinsert}%
        }%
      }%
    }%
  }%
  {%
    \glscustomtext\glsinsert
  }%
}
\newcommand*{\glsgenacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglsused\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\glssentencecase\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\glssentencecase\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\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}%
  \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@dosubst{%
        \noexpand\DTLsubstituteall\noexpand\gls@xdystring
        {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
      }%
      \gls@dosubst
    }%
  \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}%
    \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@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
}
\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}%
    \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\glsifplural\@secondoftwo
    \let\glscapscase\@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*{\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\glscapscase\@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\glscapscase\@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*{\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\glscapscase\@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\glscapscase\@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*{\GLSpl}{\@gls@hyp@opt\@GLSpl}
\glsmfublocker{\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\glscapscase\@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\glscapscase\@firstofthree
    \def\glscustomtext{#3}%
    \def\glsinsert{}%
    \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][]{%
  \@glsdisp[#1]{#2}{\glssentencecase{#3}}%
}
\glsmfuaddmap{\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}%
  }%
  \glspostlinkhook
}
\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*{\GLStext}{\@gls@hyp@opt\@GLStext}
\glsmfublocker{\GLStext}
\newcommand*{\@GLStext}[2][]{%
  \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2}[]}}
\def\@GLStext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrytext{#2}#3}}%
}
\newrobustcmd*{\Glstext}{\@gls@hyp@opt\@Glstext}
\glsmfuaddmap{\glstext}{\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}
\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}{\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}{\glsuppercase{\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}{\glsuppercase{\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*{\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*{\Glssymbol}{\@gls@hyp@opt\@Glssymbol}
\glsmfuaddmap{\glssymbol}{\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*{\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}{\glsuppercase{\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*{\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}{\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@\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\glscapscase\@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@\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{\glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname 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\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}
\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\glscapscase\@secondofthree
    \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}
\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\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \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\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\glssentencecase\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]{%
  \glstexorpdfstring{\@Gls@entry@field{#1}{#2}}%
  {\MFUsentencecase{\@gls@entry@field{#1}{#2}}}%
}
\ifdef\texorpdfstring
{\newcommand{\glstexorpdfstring}{\texorpdfstring}}
{\newcommand{\glstexorpdfstring}[2]{#1}}
\newcommand*{\glsentryname}[1]{\@gls@entry@field{#1}{name}}
\newcommand*{\Glsentryname}[1]{%
  \glstexorpdfstring{\protect\@Gls@entryname{#1}}%
  {\MFUsentencecase{\@gls@entry@field{#1}{name}}}%
}
```

```
\glsmfuaddmap{\glsentryname}{\Glsentryname}
\newcommand*{\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
\newrobustcmd*{\@Gls@acrentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \protected@edef\@glo@text{\@gls@entry@field{#1}{name}}%
    \expandafter\glssentencecase\expandafter{\@glo@text}%
  }%
  {%
    \@Gls@entry@field{#1}{name}%
  }%
}
\newcommand*{\@org@Gls@acrentryname}[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
  }%
  {%
    \@Gls@entry@field{#1}{name}%
  }%
}
\newcommand*{\glsentrydesc}[1]{\@gls@entry@field{#1}{desc}}
\newcommand*{\Glsentrydesc}[1]{\@Glsentryfield{#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*{\Glsentryuserv}[1]{%
  \@Glsentryfield{#1}{userv}%
}
\glsmfuaddmap{\glsentryuserv}{\Glsentryuserv}
\newcommand*{\glsentryuservi}[1]{%
  \@gls@entry@field{#1}{uservi}%
}
\newcommand*{\Glsentryuservi}[1]{%
  \@Glsentryfield{#1}{uservi}%
}
\glsmfuaddmap{\glsentryuservi}{\Glsentryuservi}
\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*{\glshyperlink}[2][\glsentrytext{\@glo@label}]{%
```

```
 \def\@glo@label{#2}%
 \@glslink{\glolinkprefix\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\glstildechar{\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]{%
    \glsnoxindywarning\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]{%
    \glsnoxindywarning\GlsSetXdyNumberGroupOrder}
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
    \@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*{\@glsminrange}{2}
\ifglsxindy
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \renewcommand*{\@glsminrange}{#1}}
\else
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \glsnoxindywarning\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\glsreseteentrylist
       \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"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
  \@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}{}
\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#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@checkescbar####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{\@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
  }%
  {%
    \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}%
     \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}{%
  \ifglsxindy
    \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@dodeflistparser{\noexpand\DeclareListParser
```

```
      {\noexpand\glsnumlistparser}{\delimN}}%
    \@gls@dodeflistparser
  \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{\glswrglossdisablelocationcmds}{%
    \string\@gls@reference
      {\csname glo@\@gls@label @type\endcsname}%
      {\@gls@label}%
      {\string\glsnoidxdisplayloc
        {\@glo@counterprefix}%
        {\@gls@counter}%
        {\@glsnumberformat}%
        {\@glslocref}%
      }%
  }%
}
\AtEndDocument{%
  \glswritefiles
}
\newcommand*{\@glswritefiles}{%
  \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\begingroup\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]{%
  \ifglsesclocations
    \@@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{\theHglsentrycounter}%
  \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\@glslocref{\theglsentrycounter}%
  \endgroup
  \@gls@checkmkidxchars\@glslocref
  \expandafter\ifx\theHglsentrycounter\theglsentrycounter\relax
    \def\@glo@counterprefix{}%
  \else
    \protected@edef\@glsHlocref{\theHglsentrycounter}%
    \@gls@checkmkidxchars\@glsHlocref
    \@xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
  \fi
  \protected@edef\@gls@label{\glsdetoklabel{#1}}%
  \@@do@@wrglossary
}
\newcommand*{\@@do@@wrglossary}{%
  \ifglsxindy
    \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
    \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^^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
      :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]{%
 \ifglshasshort{#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@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}{%
  \forallglossaries{\@@glo@type}{\printnoidxglossary[type=\@@glo@type]}%
}
\newcommand*{\@printgloss@setsort}{}
\newcommand*{\@gls@preglossaryhook}{}
\newcommand{\@printglossary}[2]{%
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{\csname @glotype@\@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@dotoctitle{\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 @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@dotoctitle
     \@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]{%
  \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{}%
  \forglsentries[#1]{\@gls@thislabel}%
  {%
    \xifinlistcs{\@gls@thislabel}{@glsref@#1}%
    {%
      \listeadd{\@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@\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@dotoctitle\relax
}
```

```
\define@key{printgloss}{toctitle}{%
 \def\glossarytoctitle{#1}%
 \let\gls@dotoctitle\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{glssubentrycounter#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*{\glsreseteentrycounter}{%
  \ifglsentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*{\glsstepentry}[1]{%
  \ifglsentrycounter
    \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]{%
 \ifglsentrycounter
   \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
 \else
   \ifglssubentrycounter
     \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
   \else
     \gls{#1}%
   \fi
 \fi
}
\newcommand*{\glsentrycounterlabel}{%
 \ifglsentrycounter
   \theglossaryentry.\space
 \fi
}
\newcommand*{\glssubentrycounterlabel}{%
 \ifglssubentrycounter
   \theglossarysubentry)\space
 \fi
}
\newcommand*{\glsentryitem}[1]{%
  \ifglsentrycounter
    \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*{\glstarget}[2]{\@glstarget{\glolinkprefix#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\glshypernumber#1{#1}%
}%
{%
  \def\glshypernumber#1{\@xp@glshypernumber{#1}}
}
\newrobustcmd{\@xp@glshypernumber}[1]{%
  \@glshypernumber#1\nohyperpage{}\@nil
}
\def\@glshypernumber#1\nohyperpage#2#3\@nil{%
  \ifx\\#1\\%
  \else
    \gls@delimR#1\delimR\delimR\\%
  \fi
  \ifx\\#2\\%
  \else
    #2%
  \fi
  \ifx\\#3\\%
  \else
    \@glshypernumber#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{\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][]{%
 \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\glsspace(#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@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\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}{\glsgenacfmt}{\glsgenentryfmt}%
}%
```

```
{%
  \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}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \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}{\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]{%
    \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
        }%
        {%
          \glsuppercase
            {\glsentrylongpl{\glslabel}\glsinsert}%
        }%
      }%
      {%
        \glscapscase
        {%
          \glsentrylong{\glslabel}\glsinsert
        }%
        {%
          \Glsentrylong{\glslabel}\glsinsert
        }%
        {%
          \glsuppercase
            {\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}{%
      \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}%
}
\newacronymstyle{dua-desc}%
{%
  \GlsUseAcrEntryDispStyle{dua}%
}%
{%
  \GlsUseAcrStyleDefs{dua}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymentry}[1]{\glsentrylong{##1}}%
  \renewcommand*{\acronymsort}[2]{##2}%
}%
\newacronymstyle{footnote}%
{%
  \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \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}{%
      \glsuppercase{\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}{%
      \glsuppercase{\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*{\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}[=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*{\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@\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 @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 glolist@#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*{\@glsnavhypertarget}[3]{%
  \protected@write\@auxout{}{\string\@gls@hypergroup{#1}{#2}}%
  \@glstarget{\glsnavhyperlinkname{#1}{#2}}{#3}%
  \expandafter\let
     \expandafter\@gls@list\csname @gls@hypergrouplist@#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@hypergrouprerun{%
      \GlossariesWarningNoLine{Navigation panel
      has changed. Rerun LaTeX}}%
  \fi
}
\let\gls@hypergrouprerun\relax
\AtEndDocument{\gls@hypergrouprerun}
\newcommand*{\@gls@hypergroup}[2]{%
 \ifcsdef{@gls@hypergrouplist@#1}%
 {\protected@csxappto{@gls@hypergrouplist@#1}{,#2}}%
 {\protected@csxdef{@gls@hypergrouplist@#1}{#2}}%
}
\newcommand*{\glsnavigation}{%
  \def\@gls@between{}%
  \ifcsundef{@gls@hypergrouplist@\currentglossary}%
  {%
    \def\@gls@list{}%
  }%
  {%
    \letcs\@gls@list{@gls@hypergrouplist@\currentglossary}%
  }%
  \@for\@gls@tmp:=\@gls@list\do{%
    \@gls@between
```

```
    \@gls@getgrouptitle{\@gls@tmp}{\@gls@grptitle}%
    \expandafter\glsnavhyperlink\expandafter{\@gls@tmp}{\@gls@grptitle}%
    \let\@gls@between\glshypernavsep
  }%
}
\newcommand*{\glshypernavsep}{\space\textbar\space}
\newcommand*{\glssymbolnav}{%
 \glsnavhyperlink{glssymbols}{\glsgetgrouptitle{glssymbols}}%
 \glshypernavsep
 \glsnavhyperlink{glsnumbers}{\glsgetgrouptitle{glsnumbers}}%
 \glshypernavsep
}
```

Change History

1.01 – 2007-05-17
General: Added range facility in format key 126
\writeist: Added spaces after \delimN and \delimR in ist file 181
1.04 – 2007-08-03
General: Added \glstextformat 114
1.05 – 2007-08-10
\glossarysection: added \@mkboth to \glossarysection 51
\gls@defglossaryentry: Changed the default value of the sort key to just the value of the name key 95
1.07 – 2007-09-13
\@gls@link: fixed bug caused by \theglsentrycounter setting the page number too soon . 123
\glsadd: fixed bug caused by \theglsentrycounter setting the page number too soon . 177
1.08 – 2007-10-13
General: Added babel support . 45
altlistgroup: changed altlistgroup style to use \glsgetgrouptitle 280
listgroup: changed listgroup style to use \glsgetgrouptitle 279
1.1 – 2008-02-22
General: amsgen now loaded (\new@ifnextchar needed) . . 6
\@glossarysection: numbered sections and auto label added 52
\@gls@tmpb: changed \toksdef to \newtoks 128
\@gls@toc: numberline added . . 54
\@p@glossarysection: numbered sections and auto label added 53
numberedsection: numberedsection package option added 10
numberline: numberline option added 10
\setglossarysection: new . . . 52
translate: translate option added 30
1.12 – 2008-03-08
General: added check for \hypertarget separate to \hyperlink (memoir defines \hyperlink but not \hypertarget) 136
\@GLSpl: now uses \glsentrydescplural and \glsentrysymbolplural instead of \glsentrydesc and \glsentrysymbol 141
\@Glspl@: now uses \glsentrydescplural and \glsentrysymbolplural instead of \glsentrydesc and \glsentrysymbol 140
\@glspl@: now uses \glsentrydescplural and \glsentrysymbolplural instead of \glsentrydesc and \glsentrysymbol 140
descriptionplural: new 75
\gls@defglossaryentry: Changed default first plural to be first key with s appended (was text key with s appended) 95
descriptionplural support added 95
symbolplural support added . 95
\Glsentrydescplural: New . . . 169
\glsentrydescplural: New . . . 169
\Glsentrysymbolplural: New . 170
\glsentrysymbolplural: New . 170
symbolplural: new 77
1.13 – 2008-05-10
General: fixed bug that ignored 3rd parameter 143–152
\ACRfullpl: new 246
\Acrfullpl: new 245
\acrfullpl: new 245
\acrpluralsuffix: New 242
\gls@defglossaryentry: Changed default first value . 95

Changed default firstplural value 95
Removed restriction on only using \newglossaryentry in the preamble 100
\newacronym: Removed restriction on only using \newacronym in the preamble 242
1.14 – 2008-06-17
General: added nonumberlist key to \printglossary 229
added numberedsection key to \printglossary 228
\@gls@hypergroup: new 274
\firstacronymfont: new 246
\glsautoprefix: new 10
\glsnavhyperlink: changed \edef to \protected@edef 271
\glsnavhypertarget: added write to aux file 273
\glsnavigation: changed to only use labels for groups that are present 274
1.15 – 2008-08-15
\@gls@link: added \glslabel . 123
\gls@defglossaryentry: check for \@glo@first in description 99
check for \@glo@text in symbol 99
\gls@hypergrouprerun: new . . 274
\glsnavhypertarget: added check if rerun required . . . 273
\glssettoctitle: new 44
\printglossary: changed the way the TOC title is set . . 213
1.16 – 2008-08-27
\@GLS@: Test glossary type is \acronymtype in addition to checking if footnote option has been used 139
\@GLSpl: Test glossary type is \acronymtype in addition to checking if footnote option has been used 141
\@Gls@: Test glossary type is \acronymtype in addition to checking if footnote option has been used 138
\@Glspl@: Test glossary type is \acronymtype in addition to checking if footnote option has been used 141
\@gls@: Test glossary type is \acronymtype in addition to checking if footnote option has been used 137
\@glsdisp: Test glossary type is \acronymtype in addition to checking if footnote option has been used 142
\@glspl@: Test glossary type is \acronymtype in addition to checking if footnote option has been used 140
\@glstarget: raised the hypertarget so the target text doesn't scroll off the top of the page 136
\gls@defglossaryentry: Changed def to let 95
1.17 – 2008-12-26
General: added xindy support . 32
\@@do@esc@wrglossary: new . . 206
\@do@seeglossary: new 210
\@glo@storeentry: new 102
\@gls@glossary: changed definition to use \index instead of \@index 202
\@glsdefaultplural: new 80
\@glsdefaultsort: new 81
\@glshypernumber: new 238
\@glsnoname: new 80
\@glsnonextpages: new 229
\gls@defglossaryentry: added nonumberlist key 96
added parent key 95
added see key 95
Stored main part of entry format when entry is defined 100
\gls@suffixF: new 49
\gls@suffixFF: new 49
\gls@wrglossary: modified to allow for xindy support . . . 203
\glshyperlink: new 177
\glshypernumber: modified to allow material to be attached to location 238
\glsnavhyperlink: replaced \hyperlink to \@glslink . 271

\glsnavhypertarget: replaced \hypertarget with \@glstarget 273
\glssee: new 211
\glsseeformat: new 211
\glsSetSuffixF: new 49
\glsSetSuffixFF: new 49
\ifglsxindy: new 32
\istfilename: added xindy support 48
\newglossarystyle: made \newglossarystyle long . . 237
nonumberlist: new 78
\nopostdesc: new 47
order: order package option added 32
parent: new 78
\printglossary: added check to determine if \printglossary is already defined 213
added print language to aux file 213
see: new 77
\writeist: added xindy support 181
1.18 – 2009-01-14
\@gls@loadlist: new 13
\@gls@loadlong: new 12
\@gls@loadsuper: new 12
\@gls@loadtree: new 13
\gls@defglossaryentry: Changed default value of sort to \@glsdefaultsort 95
moved sort sanitization to \newglossaryentry 99
\glstarget: new 232
nolist: new 13
nolong: new 12
nostyles: new 13
nosuper: new 13
notree: new 13
\oldacronym: new 241
sort: moved sanitization to \newglossaryentry 76
1.19 – 2009-03-02
\glsclearpage: new 54
\glsdisp: new 142
1.46 – 2021-09-20
General: changed \edef to \protected@edef 373
\@printglossary: changed \xdef to \protected@xdef 216
\gls@defglossaryentry: changed \xdef to \protected@xdef 97, 100
\glsaddprotectedpagefmt: changed \eappto to \protected@eappto 204
2.01 – 2009 May 30
\@gls@link: moved \@do@wrglossary before term is displayed to prevent unwanted whatsit 124
\forallglossaries: replaced \ifthenelse with \ifx 63
\forglsentries: replaced \ifthenelse with \ifx 64
\glsdefmain: new 19
\glsdescwidth: changed \linewidth to \hsize . 282, 305
\glslistdottedwidth: changed \linewidth to \hsize 281
\glspagelistwidth: changed \linewidth to \hsize . 282, 305
nomain: added nomain package option 20
\writeist: removed item_02 - no such makeindex key . . . 185
2.02 – 2007-07-13
\@printglossary: suppressed warning globally rather than locally 216
2.02 – 2009-07-13
\glossarysection: changed \@mkboth to \glossarymark 51
\glsglossarymark: New 51
2.03 – 2009-09-23
\@GLS@: Added check for hyperfirst 139
\@GLSpl: Added check for hyperfirst 141
\@Gls@: Added check for hyperfirst 138
\@Glspl@: Added check for hyperfirst 141
\@gls@: Added check for hyperfirst 137
\@gls@@link: new 122
\@gls@link: added \leavevmode 123

Moved entry existence check to avoid duplicate code 123
\@glsdisp: Added check for hyperfirst 142
\@glspl@: Added check for hyperfirst 140
\glsglossarymark: Added check to see if it's already defined . 51
hyperfirst: new 31
2.04 – 2009-11-10
\@GLS@: Changed test to check if glossary type has been identified as a list of acronyms 139
\@GLSpl: Changed test to check if glossary type has been identified as a list of acronyms 141
\@Gls@: Changed test to check if glossary type has been identified as a list of acronyms 138
\@Glspl@: Changed test to check if glossary type has been identified as a list of acronyms 141
\@glossaryentryfield: new . . 101
\@glossarysubentryfield: new 101
\@gls@: Changed test to check if glossary type has been identified as a list of acronyms 137
\@glsacronymlists: new 21
\@glsdisp: Changed test to check if glossary type has been identified as a list of acronyms 142
\@glspl@: Changed test to check if glossary type has been identified as a list of acronyms 140
\@newglossaryentryposthook: new 101
\@newglossaryentryprehook: new 101
acronymlists: new 22
\DeclareAcronymList: new . . . 21
\DefineAcronymSynonyms: new . 259
\gls@defglossaryentry: added user1-6 keys 96
\glsadd: fixed bug that ignored counter 177
\Glsentryuseri: new 172
\glsentryuseri: new 172
\Glsentryuserii: new 172
\glsentryuserii: new 172
\Glsentryuseriii: new 173
\glsentryuseriii: new 173
\Glsentryuseriv: new 173
\glsentryuseriv: new 173
\Glsentryuserv: new 173
\glsentryuserv: new 173
\Glsentryuservi: new 174
\glsentryuservi: new 173
\ns@newglossary: added check to determine if \gls@⟨type⟩@display and \gls@⟨type⟩@displayfirst have been defined. 73
\SetAcronymLists: new 22
\SetDefaultAcronymDisplayStyle: new 261
\SetDefaultAcronymStyle: new 261
\SetDescriptionAcronymDisplayStyle: new 262
\SetDescriptionDUAAcronymDisplayStyle: new 262
\SetDescriptionFootnoteAcronymDisplayStyle: new 262
\SetDUADisplayStyle: new . . . 263
\SetFootnoteAcronymDisplayStyle: new 262
\SetSmallAcronymDisplayStyle: new 263
2.05 – 2010-02-06
\@glsdisp: Added closing brace. Patch provided by Sergiu Dotenco 142
Removed spurious brace. Patch provided by Sergiu Dotenco 142
\writeist: Added \string before opening and closing braces. Patch provided by Segiu Dotenco 186
2.06 – 2010-06-14
\altnewglossary: new 73
\CustomAcronymFields: new . . 264
\CustomNewAcronymDef: new . . 264
\SetCustomDisplayStyle: new . 264
\SetCustomStyle: new 264

2.07 – 2010-07-10
General: glsadd format key stored in \@glsnumberformat (was mistakenly stored in \@glo@format) 177
3.0 – 2010-07-12
\@makeglossary: Added check for savewrites 190
\gls@wrglossary: modified to take into account savewrites 203
3.0 – 2010/03/31
\@set@glo@numformat: added 4th argument 126
3.0 – 2011-04-02
General: added prefix to hyperlink 239
etoolbox now loaded 6
replaced \@ifundefined with \ifcsundef . . 43, 46, 119, 227
\@@do@esc@wrglossary: added check for hyper location prefix 208
modified to use new format . 206
\@@glossarysec: replaced \@ifundefined with \ifcsundef 10
\@do@seeglossary: Sanitize and escape cross-referencing information 210
\@gls@counterwithin: new . . . 15
\@gls@ifinlist: new 55
\@gls@link: added \@gls@saveentrycounter . 124
added \@gls@setsort 124
\@gls@saveentrycounter: new . 125
\@gls@setupsort@def: new . . . 17
\@gls@setupsort@standard: new 16
\@gls@setupsort@use: new . . . 17
\@gls@xdy@locationlist: new . 58
\@glslink: replaced \@ifundefined with \ifcsundef 136
\@glsnextpages: new 229
\@print@glossary: replaced \@ifundefined with \ifcsundef 217
\@printglossary: added \currentglossary 215
added \glsnextpages 215
make toctitle default to title . 215
\@xdyattributelist: new 54
\acrfootnote: new 261
\ACRfull: added starred version 245
\Acrfull: added starred version 244
\acrfull: added starred version 244
\ACRfullpl: added starred version 246
\Acrfullpl: added starred version 245
\acrfullpl: added starred version 245
\acrlinkfootnote: new 261
\acrnolinkfootnote: new 262
compatible-2.07: compatible-2.07 option added 38
counter: replaced \@ifundefined with \ifcsundef 77
counterwithin: new 14
entrycounter: new 14
glossaryentry: new 14
\glossarysection: replaced \@ifundefined with \ifcsundef 51
\glossarystyle: replaced \@ifundefined with \ifcsundef 237
glossarysubentry: new 15
\gls@codepage: replaced \@ifundefined with \ifcsundef 33
\gls@defglossaryentry: added \@gls@defsort 99
added short and long keys . . 96
replaced \@ifundefined with \ifcsundef 96
\gls@doclearpage: replaced \@ifundefined with \ifcsundef 53
\glsadd: added \@gls@saveentrycounter . 177
\GlsAddXdyCounters: new 55
\glsentrycounterlabel: new . 231
\glsentryitem: new 231
\Glsentrylong: new 174
\glsentrylong: new 174
\Glsentrylongpl: new 174
\glsentrylongpl: new 174
\Glsentryshort: new 174
\glsentryshort: new 174
\Glsentryshortpl: new 174

\glsentryshortpl: new 174
\glsgetgrouptitle: replaced \@ifundefined with \ifcsundef 235
\glsglossarymark: replaced \@ifundefined with \ifcsundef 51
\glshyperlink: changed default from \glsentryname to \glsentrytext 177
\glshypernumber: replaced \@ifundefined with \ifcsundef 238
\glsnumberformat: replaced \@ifundefined with \ifcsundef 49
\glsrefentry: new 231
\glsresetsubentrycounter: new 230
\glsseeitem: hyperlink uses \glsseeitemformat instead of \glsentryname 212
\glsseeitemformat: new 212
\glssortnumberfmt: new 16
\glsstepentry: new 230
\glsstepsubentry: new 230
\glssubentrycounterlabel: new 231
\glssubentryitem: new 231
\ifglossaryexists: replaced \@ifundefined with \ifcsundef 64
\ifglsentryexists: replaced \@ifundefined with \ifcsundef 65
\istfile: deprecated 200
long: new 80
longplural: new 80
\newglossaryentry: replaced \DeclareRobustCommand with \newrobustcmd 83
\newglossarystyle: replaced \@ifundefined with \ifcsundef 237
nonumberlist: now boolean . . . 78
\ns@newglossary: added \@gls@defsortcount 73
replaced \@ifundefined with \ifcsundef 73
\oldacronym: replaced \@ifundefined with \ifcsundef 241
\printglossary: replaced \@ifundefined with \ifcsundef 213
savewrites: new 38
see: added \@glo@seeautonumberlist . 77
seeautonumberlist: new 12
\SetDescriptionFootnoteAcronymDisplayStyle: expanded options link options 262
\setentrycounter: added optional argument 236
short: new 80
shortplural: new 80
\showacronymlists: new 269
\showglocounter: new 266
\showglodesc: new 267
\showglodescplural: new 268
\showglofirst: new 265
\showglofirstpl: new 266
\showgloflag: new 269
\showgloindex: new 269
\showglolevel: new 265
\showgloname: new 267
\showgloparent: new 265
\showgloplural: new 265
\showglosort: new 268
\showglossaries: new 269
\showglossarycounter: new . . 270
\showglossaryentries: new . . 270
\showglossaryin: new 270
\showglossaryout: new 270
\showglossarytitle: new 270
\showglosymbol: new 268
\showglosymbolplural: new . . 268
\showglotext: new 265
\showglotype: new 266
\showglouseri: new 266
\showglouserii: new 266
\showglouseriii: new 267
\showglouseriv: new 267
\showglouserv: new 267
\showglouservi: new 267
sort: new 15
subentrycounter: new 15

theglossary: replaced \@ifundefined with \ifcsundefined 232
\writeist: added xindy-only macro definitions to glossary open tag 183
modified to support new format 181
3.01 – 2011-04-12
General: made robust 138
\@glswritefiles: added check for empty glossaries 201
\ACRfull: made robust 245
\Acrfull: made robust 244
\acrfull: made robust 244
\acrfullformat: removed \acronymfont as it should already be set in the second argument. 243
\ACRfullpl: made robust 246
\Acrfullpl: made robust 245
\acrfullpl: made robust 245
\ACRlong: made robust 163
\Acrlong: made robust 162
\acrlong: made robust 161
\ACRlongpl: made robust 165
\Acrlongpl: made robust 164
\acrlongpl: made robust 164
\ACRshort: made robust 159
\Acrshort: made robust 158
\acrshort: made robust 158
\ACRshortpl: made robust 161
\Acrshortpl: made robust 160
\acrshortpl: made robust 160
\Gls: made robust 138
\glsadd: made robust 177
\glsaddall: made robust 178
\GLSdesc: made robust 148
\Glsdesc: made robust 148
\glsdesc: made robust 148
\GLSdescplural: made robust . 149
\Glsdescplural: made robust . 149
\glsdescplural: made robust . 149
\glsfirst: made robust 144
\GLSfirstplural: made robust 146
\Glsfirstplural: made robust 146
\glsfirstplural: made robust 146
\glslink: made robust 122
\GLSname: made robust 147
\Glsname: made robust 147
\glsname: made robust 147
\GLSpl: made robust 141
\Glspl: made robust 140
\glspl: made robust 139
\GLSplural: made robust 145
\GLSsymbol: made robust 150
\Glssymbol: made robust 150
\glssymbol: made robust 150
\GLSsymbolplural: made robust 151
\Glssymbolplural: made robust 151
\glssymbolplural: made robust 151
\Glstext: made robust 143
\glstext: made robust 143
\GLSuseri: made robust 152
\Glsuseri: made robust 152
\glsuseri: made robust 152
\GLSuserii: made robust 153
\Glsuserii: made robust 153
\glsuserii: made robust 153
\GLSuseriii: made robust 154
\Glsuseriii: made robust 154
\glsuseriii: made robust 154
\GLSuseriv: made robust 155
\Glsuseriv: made robust 155
\glsuseriv: made robust 155
\GLSuserv: made robust 156
\Glsuserv: made robust 156
\glsuserv: made robust 156
\GLSuservi: made robust 157
\Glsuservi: made robust 157
\glsuservi: made robust 157
3.02 – 2012-05-19
\glsnumlistlastsep: new 176
\glsnumlistsep: new 176
3.02 – 2012-05-21
\@@do@@wrglossary: changed \@glslocref to \theglsentrycounter 209
\@do@wrglossary: changed \@do@wr@glossary to test for indexonlyfirst option; put old \@do@wr@glossary code into \@@do@wrglossary 203
\@gls@missingnumberlist: new 81
\@glswritefiles: added check for existence of token in case \makeglossaries has been omitted 201
\@printglossary: add a way to fetch current entry label . . 216

\gls@defglossaryentry: added numberlist element 99
\gls@save@numberlist: new . . 212
\gls@wrglossary: added check for glossary file defined . . . 203
\glsdisplaynumberlist: new . 175
\glsentrycounter: set default value 125
\Glsentryfull: fixed bug (replaced \glsentryshortpl with \glsentryshort) 175
\glsentryfullpl: fixed bug (replaced \glsentryshort with \glsentryshortpl) . . 175
\glsentrynumberlist: new . . . 175
\glsmoveentry: new 101
\glsresetentrycounter: new . 230
\ifglshaschildren: new 67
\ifglshasparent: new 67
indexonlyfirst: new 31
\makeglossaries: added list parser 195
\renewglossarystyle: new . . . 237
savenumberlist: new 12
\showglossaryentries: fixed misspelt command 270
\SmallNewAcronymDef: fixed broken short and long plural 263
ucmark: new 14
3.03 – 2012/09/21
General: allow title to set toctitle 227
\@gls@sanitizesort: new 25
\@gls@setupsort@standard: used \@gls@sanitizesort 16
\@printglossary: allow title to override default toctitle . . . 214
altlongragged4col: added check for glsnogroupskip 297
altsuperragged4col: added check for glsnogroupskip . . 316
alttree: added check for glsnogroupskip 326
\glsinlinedescformat: new . . 277
\glsinlineemptydescformat: new 277
\glsinlinenameformat: new . . 277
\glsinlinepostchild: new . . . 277
\glsinlinesubdescformat: new 277
\glsinlinesubnameformat: new 277
\glspostinline: replaced "." with \glspostdescription 277
index: added check for glsnogroupskip 320
list: added check for glsnogroupskip 279
long: added check for glsnogroupskip 283
long3col: added check for glsnogroupskip 284
long4col: added check for glsnogroupskip 286
longragged: added check for glsnogroupskip 294
longragged3col: added check for glsnogroupskip 295
nogroupskip: new 14
nopostdot: new 13
super: added check for glsnogroupskip 305
super3col: added check for glsnogroupskip 307
super4col: added check for glsnogroupskip 309
superragged: added check for glsnogroupskip 313
superragged3col: added check for glsnogroupskip 315
tree: added check for glsnogroupskip 322
treenoname: added check for glsnogroupskip 323
3.04 – 2012-11-11
altlist: replaced \newline with paragraph break 280
3.04 – 2012-11-18
General: Added check for doc package 6
added datatool-base as a required package 6
added local key 121
\@@do@@wrglossary: changed \theglsentrycounter back to \@glslocref 209
\@@do@esc@wrglossary: modified to compensate for possible incorrect page number 207
\@gls@escbsdq: unsanitize \gls@numberpage, \gls@alphpage,

\gls@Alphpage and \gls@romanpage 127
\@print@glossary: Moved aux write to end of document to prevent unwanted whatsit occurring here. 217
\gls@Alphpage: new 204
\gls@alphpage: new 204
\gls@disablepagerefexpansion: new 203
\gls@numberpage: new 204
\gls@protected@pagefmts: new 203
\gls@romanpage: new 204
\glsdefmain: added check for doc package 19
\glsorg@endtheglossary: new . . 6
\glsorg@theglossary: new 6
\PrintChanges: new 6
3.05 – 2013-04-21
\@@do@esc@wrglossary: add Roman case. Fixed bugs in the else statements 207
\@gls@link: added check for "nohypertypes" 124
\gls@protected@pagefmts: added Roman to list 203
\gls@Romanpage: new 204
\glsgetgrouplabel: fixed bug (typo in \equal) 236
\glsmcols: new 299
mcolalttree: replaced '2' with \glsmcols 303
mcolindex: replaced '2' with \glsmcols 299
mcolindexspannav: replaced '2' with \glsmcols 300
mcoltree: replaced '2' with \glsmcols 300
mcoltreenoname: replaced '2' with \glsmcols 302
mcoltreespannav: replaced '2' with \glsmcols 301
\nopostdesc: made robust 47
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 \languagename 33
\findrootlanguage: Obsoleted . 62
3.07 – 2013-07-05
\@gls@link: fixed bug that failed to find entry in list 124
\glossarypreamble: modified to work with \setglossarypreamble 50
\gls@doclearpage: added check for openright 53
\glspostdescription: Added spacefactor code 13
\GlsSetXdyCodePage: Added check for fontspec 62
\SetDescriptionAcronymDisplayStyle: now using \glsdoparenifnotempty . . 262
\setglossarypreamble: new . . 50
3.08a – 2013-08-30
altlist: updated altlist style to use \glossentry and \subglossentry 280
inline: updated inline style to use \glossentry and \subglossentry 275
list: updated list style to use \glossentry and \subglossentry 279
listdotted: updated listdotted style to use \glossentry and \subglossentry 281
3.08a – 2013-09-28
General: added nogroupskip key to \printglossary 228
removed definition of \@glossaryentryfield . . . 362
removed definition of \@glossarysubentryfield 362
\@glo@storeentry: no longer need to check for special characters in any of the fields other than sort 102
updated for \glossentry . . . 102
\@glossaryentryfield: switched to \glossentry 101
\@glossarysubentryfield: switched to \subglossentry 101
\@gls@sanitizedesc: deprecated 24
altlongragged4col: updated to use \glossentry and \subglossentry 297

alttree: updated to use \glossentry and \subglossentry 325
\compatibleglossentry: new . 232
\compatiblesubglossentry: new 234
\glossaryentryfield: deprecated 234
\glossarystyle: deprecated . . 237
\glossentry: new 232
\Glossentrydesc: new 233
\glossentrydesc: new 233
\Glossentryname: new 233
\glossentryname: new 232
\Glossentrysymbol: new 233
\glossentrysymbol: new 233
\gls@assign@desc@field: new . 24
\gls@assign@descplural@field: new 25
\gls@assign@field: new 83
\gls@ifnotmeasuring: new . . . 104
\glsaddallunused: new 178
\glsexpandfields: new 83
\glsnoexpandfields: new 83
\glssee: made robust 211
\glsseeformat: made robust . . 211
\glsseeitem: made robust 212
\glsseelist: made robust 211
\ifglsdescsuppressed: new . . 68
\ifglshasdesc: new 67
\ifglshassymbol: new 68
index: added paragraph break at end of environment 319
updated to use \glossentry and \subglossentry 320
long: updated to use \glossentry and \subglossentry 282
longragged: updated to use \glossentry and \subglossentry 294
longragged3col: updated to use \glossentry and \subglossentry 295
\setglossarystyle: new 236
\setglossentrycompatibility: new 234
\subglossentry: new 234
superragged: updated to use \glossentry and \subglossentry 312
tree: updated to use \glossentry and \subglossentry 321
3.09a – 2013-10-09
\@gls@assign@symbolplural@field: new 25
\@gls@default@value: new . . . 76
\Glsentrydesc: made robust . . 169
\Glsentrydescplural: made robust 169
\Glsentryfirst: made robust . 171
\Glsentryfirstplural: made robust 171
\Glsentryfull: made robust . . 175
\Glsentryfullpl: made robust 175
\Glsentrylong: made robust . . 174
\Glsentrylongpl: made robust 174
\Glsentryname: made robust . . 167
\Glsentryplural: made robust 170
\Glsentryshort: made robust . 174
\Glsentryshortpl: made robust 174
\Glsentrysymbol: made robust 170
\Glsentrysymbolplural: made robust 170
\Glsentrytext: made robust . . 170
\Glsentryuseri: made robust . 172
\Glsentryuserii: made robust 172
\Glsentryuseriii: made robust 173
\Glsentryuseriv: made robust 173
\Glsentryuserv: made robust . 173
\Glsentryuservi: made robust 174
\glstextup: new 242
\ifglshassymbol: changed test to check for \@gls@default@symbol 68
3.10a – 2013-09-28
\gls@assign@type@field: new . 24
3.10a – 2013-10-13
\@gls@keymap: new 86
\@gls@provide@newglossary: new 71
\@gls@writedef: new 86
\@glsdefaultplural: Obsolete . 80
\@glsnodesc: new 80
\@print@glossary: Added providecommand code to aux file 217

\gls@defglossaryentry: Changed to using \@gls@default@value 95
new 95
\glswritedefhook: new 94
\makeglossaries: Added providecommand code to aux file 194
\new@glossaryentry: new 84
\ns@newglossary: added \@gls@provide@newglossary 73
3.11a – 2013-10-15
General: added \glslabel, \glsifplural, \glscapscase, \glsinsert and \glscustomtext 158–165
changed to just use \Glsentrydescplural 149
changed to just use \glsentrydescplural . 149, 150
changed to just use \Glsentrydesc 148
changed to just use \glsentrydesc 148, 149
changed to just use \Glsentryfirstplural . . . 146
changed to just use \glsentryfirstplural 146, 147
changed to just use \Glsentryfirst 144
changed to just use \glsentryfirst 144, 145
changed to just use \Glsentryname 147
changed to just use \glsentryname 147, 148
changed to just use \Glsentryplural 145
changed to just use \glsentryplural 145, 146
changed to just use \Glsentrysymbolplural . . 151
changed to just use \glsentrysymbolplural 151, 152
changed to just use \Glsentrysymbol 150
changed to just use \glsentrysymbol 150, 151
Changed to just use \Glsentrytext 144
changed to just use \glsentrytext 143
changed to just use \Glsentryuseriii 154
changed to just use \glsentryuseriii . . . 154, 155
changed to just use \Glsentryuserii 153
changed to just use \glsentryuserii 153, 154
changed to just use \Glsentryuseriv 155
changed to just use \glsentryuseriv 155, 156
changed to just use \Glsentryuseri 152
changed to just use \glsentryuseri 152, 153
changed to just use \Glsentryuservi 157
changed to just use \glsentryuservi 157, 158
changed to just use \Glsentryuserv 156
changed to just use \glsentryuserv 156, 157
Now requires textcase 5
\@ACRlong: added \glslabel, \glsifplural, \glscapscase, \glsinsert and \glscustomtext 357
\@ACRshort: added \glslabel, \glsifplural, \glscapscase, \glsinsert and \glscustomtext 356
\@Acrlong: added \glslabel, \glsifplural, \glscapscase, \glsinsert and \glscustomtext 357
\@Acrshort: added \glslabel, \glsifplural, \glscapscase, \glsinsert and \glscustomtext 355
\@GLS@: add \glslabel, \glsifplural, \glscapscase, \glscustomtext and \glsinsert 139
change to using \glsentryfmt style commands 139

removed \MakeUppercase (now moved to \glsentryfmt) . . 139
\@GLSpl: add \glslabel, \glsifplural, \glscapscase, \glscustomtext and \glsinsert 141
change to using \glsentryfmt style commands 141
removed \MakeUppercase as now dealt with in \glsentryfmt 141
\@Gls@: add \glsifplural, \glscapscase, \glscustomtext and \glsinsert 138
change to using \glsentryfmt style commands 138
removed \makefirstuc (now dealt with in \glsentryfmt) 138
\@Glspl@: add \glsifplural, \glscapscase, \glscustomtext and \glsinsert 140
change to using \glsentryfmt style commands 140
removed \makefirstuc (now dealt with in \glsentryfmt) 141
\@acrlong: added \glslabel, \glsifplural, \glscapscase, \glsinsert and \glscustomtext 356
\@acrshort: added \glslabel, \glsifplural, \glscapscase, \glsinsert and \glscustomtext 355
\@gls@: add \glslabel, \glsifplural, \glscapscase, \glscustomtext and \glsinsert 137
change to using \glsentryfmt style commands 137
\@gls@noexpand@fields: Fixed bug expand replaced with noexpand 81
\@glsdisp: add \glslabel, \glsifplural, \glscapscase, \glscustomtext and \glsinsert 142
change to using \glsentryfmt style commands 142
\@glspl@: add \glslabel, \glsifplural, \glscapscase, \glscustomtext and \glsinsert 140
change to using \glsentryfmt style commands 140
acronymlists: replaced \@addtoacronymlists with \DeclareAcronymList 22
compatible-3.07: new 38
\defglsdisplay: obsoleted 119
\defglsdisplayfirst: obsoleted 119
\defglsentryfmt: new 71
\forglsentries: replaced \ifx with \ifdefempty 64
\gls@assign@desc: new 94
\gls@defglossaryentry: Fixed default counter if none supplied 98
\gls@doentryfmt: new 71
\glsdisplay: obsoleted 119
\glsdisplayfirst: obsoleted . . 119
\glsgenentryfmt: new 114
\glsgetgrouptitle: Added check in case non-Latin alphabet in use 235
\glsglossarymark: replaced \MakeUppercase with \mfirstucMakeUppercase . . 51
\glsnavigation: switched to using \@gls@getgrouptitle 274
\ifglshasdesc: replaced \ifdefempty with \ifcsempty 67
\ifglshaslong: new 68
\ifglshasshort: new 68
\ifglshassymbol: replaced \ifdefempty with \ifcsempty 68
\ifglsused: replaced \ifthenelse with \ifbool . 65
\longnewglossaryentry: new . 94
\ns@newglossary: replaced \glsdisplay and \glsdisplayfirst with \glsentryfmt 73
numbers: new 40
\SetCustomDisplayStyle: updated to use \defglsentryfmt 264

\SetDefaultAcronymDisplayStyle: changed to use \defglsentryfmt 261
\SetDescriptionAcronymDisplayStyle: updated to use \defglsentryfmt 262
\SetDescriptionDUAAcronymDisplayStyle: updated to use \defglsentryfmt 262
\SetDescriptionFootnoteAcronymDisplayStyle: updated to use \defglsentryfmt 262
\SetDUADisplayStyle: updated to use \defglsentryfmt . . 263
\SetFootnoteAcronymDisplayStyle: updated to use \defglsentryfmt 262
\SetSmallAcronymDisplayStyle: updated to use \defglsentryfmt 263
\setupglossaries: new 42
\showglolong: new 268
\showgloshort: new 268
symbols: new 40
3.12a – 2013-10-16
\gls@defglossaryentry: added \glslabel 95
\glsaddkey: new 88
3.13a – 2013-11-05
\@gls@assign@symbol@field: changed to use \glssetnoexpandfield 25
\@gls@assign@symbolplural@field: changed to use \glssetnoexpandfield 25
\@gls@link: removed \relax . . 124
\@gls@notranslatorhook: new . 29
\@gls@setupsort@standard: moved \@gls@santizesort to \glsprestandardsort 16
altsuper4colheader: switched to \tabularnewline 310
altsuper4colheaderborder: switched to \tabularnewline 311
\glossarysection: changed \glossarymark to \glsglossarymark 51
\gls@assign@desc@field: changed to use \glssetnoexpandfield 24
\gls@assign@descplural@field: changed to use \glssetnoexpandfield 25
\gls@assign@name@field: changed to use \glssetnoexpandfield 25
\gls@assign@type@field: changed to use \glssetexpandfield 24
\gls@checkseeallowed: new . . 77
\glsaddallunused: set default to \@glo@types 178
\Glsentryfull: changed to use \acrfullformat 175
\glsentryfull: changed to use \acrfullformat 175
\Glsentryfullpl: changed to use \acrfullformat 175
\glsentryfullpl: changed to use \acrfullformat 175
\glsglossarymark: renamed \glossarymark to \glsglossarymark to avoid conflict with memoir 51
\glsprestandardsort: new . . . 15
\glssetexpandfield: new 24
\glssetnoexpandfield: new . . 24
long: switched to \tabularnewline 282, 283
long3col: switched to \tabularnewline 284
long3colheader: switched to \tabularnewline 285
long3colheaderborder: switched to \tabularnewline 285
long4col: switched to \tabularnewline 285
long4colheader: switched to \tabularnewline 286
longheader: switched to \tabularnewline 283
longheaderborder: switched to \tabularnewline 283
see: added \gls@checkseeallowed 77
super: switched to \tabularnewline 305

super3col: switched to \tabularnewline 307
super3colheader: switched to \tabularnewline 308
super4col: switched to \tabularnewline 309
super4colheader: switched to \tabularnewline 309
super4colheaderborder: switched to \tabularnewline 310
superheader: switched to \tabularnewline 306
superheaderborder: switched to \tabularnewline 306
ucmark: added check for memoir 14
3.14a – 2013-11-12
General: new 329
\@glswritefiles: renamed \glswritefiles to \@glswritefiles and used "savewrites" option to set \glswritefiles 201
acronyms: new 20
compatible-2.07: added check for 2.07 before setting 3.07 compatibility 38
\gls@defglossaryentry: added check for existence of default glossary 96
set the default for firstplural to be the value of plural 98
\longprovideglossaryentry: new 95
notranslate: new 30
\provideglossaryentry: new . 83
xindygloss: new 34
4.0 – 2013-11-14
\gls@defglossaryentry: added check for first key 98
super: fixed typo in \subglossentry (\glossentrydesc) 305
4.01 – 2013-11-16
General: fixed non-value options so that they can be passed to document class 11
4.02 – 2013-12-05
General: Removed \acronymfont 162–165
\@acrfull: now using \acrfullfmt 244
\@gls@indexdef: new 41
\@gls@numbersdef: new 40
\@gls@symbolsdef: new 40
\ACRfullfmt: new 244
\Acrfullfmt: new 243
\acrfullfmt: new 243
\ACRfullplfmt: new 244
\Acrfullplfmt: new 244
\acrfullplfmt: new 244
\acronymentry: new 248
dua: new 254
dua-desc: new 256
footnote: new 256
footnote-desc: new 258
footnote-sc: new 258
footnote-sc-desc: new 259
footnote-sm: new 258
footnote-sm-desc: new 259
\Genacrfullformat: new 118
\genacrfullformat: new 118
\GenericAcronymFields: new . 248
\Genplacrfullformat: new . . . 119
\genplacrfullformat: new . . . 119
\Glsentryfull: bug fix: added missing \acronymfont 175
\glsentryfull: bug fix: added missing \acronymfont 175
\Glsentryfullpl: bug fix: added missing \acronymfont 175
\glsentryfullpl: bug fix: added missing \acronymfont 175
\glsgenacfmt: new 116
\GlsUseAcrEntryDispStyle: new 249
\GlsUseAcrStyleDefs: new . . . 249
index: new 40
long-sc-short: new 251
long-sc-short-desc: new 253
long-short: new 249
long-short-desc: new 252
long-sm-short: new 252
long-sm-short-desc: new 253
\newacronymstyle: new 249
numberedsection: added nameref option 10
sanitize: fixed bug that caused an error here 29
sc-short-long: new 252

sc-short-long-desc: new 254
\setacronymstyle: new 248
\SetGenericNewAcronym: new . 246
short-long: new 251
short-long-desc: new 253
sm-short-long: new 252
sm-short-long-desc: new 254
xindynoglsnumbers: new 34
4.02 – 2013-13-05
\makeglossaries: made preamble only 196
4.03 – 2014-01-17
General: changed default to \@empty instead of \relax . . 38
4.03 – 2014-01-20
General: removed \glslabel (defined in \@gls@link) . . 158
\@@do@esc@wrglossary: added \glsdetoklabel 208
\@@do@noesc@wrglossary: added \glsdetoklabel 206
\@ACRlong: removed \glslabel (defined in \@gls@link) . . 357
\@ACRshort: removed \glslabel (defined in \@gls@link) . . 356
\@Acrlong: removed \glslabel (defined in \@gls@link) . . 357
\@Acrshort: removed \glslabel (defined in \@gls@link) . . 355
\@GLS@: removed \glslabel (defined in \@gls@link) . . 139
\@GLSpl: removed \glslabel (defined in \@gls@link) . . 141
\@Gls@: removed \glslabel (defined in \@gls@link) . . 138
\@Gls@entry@field: new 166
\@Glspl@: removed \glslabel (defined in \@gls@link) . . 140
\@acrlong: removed \glslabel (defined in \@gls@link) . . 356
\@acrshort: removed \glslabel (defined in \@gls@link) . . 355
\@gls@: removed \glslabel (defined in \@gls@link) . . 137
\@gls@access@display: new . . 346
\@gls@entry@field: new 166
\@gls@fetchfield: new 87
\@gls@field@link: new 143
\@gls@link: added \glsdetoklabel 123
moved \@gls@link@opts and \@gls@link@label to \@gls@link 123
\@gls@writedef: added \glsdetoklabel 86
\@glsdisp: removed \glslabel (defined in \@gls@link) . . 142
\@glspl@: removed \glslabel (defined in \@gls@link) . . 140
\@printglossary: added \glsdetoklabel 216
\compatibleglossentry: added \glsdetoklabel 337
\compatiblesubglossentry: added \glsdetoklabel . . . 337
dua: fixed bug in \acrfullfmt . 255
fixed bug in \Acrfullplfmt . 256
fixed bug in \acrfullplfmt . 256
redefined to use accessibility information 367
dua-desc: commented spurious EOL 256
redefined to use accessibility information 369
footnote: redefined to use accessibility information . . 370
footnote-desc: redefined to use accessibility information . . 372
footnote-sc: redefined to use accessibility information . . 371
footnote-sc-desc: redefined to use accessibility information 372
footnote-sm: redefined to use accessibility information . . 372
footnote-sm-desc: redefined to use accessibility information 372
\Genacrfullformat: redefined to use accessibility information 355
\genacrfullformat: redefined to use accessibility information 354
\Genplacrfullformat: redefined to use accessibility information 355
\genplacrfullformat: redefined to use accessibility information 355
\glossentryname: added \glsdetoklabel 232
\gls@defglossaryentry: added \glsdetoklabel 95

replaced #1 with \@glo@label 96
replaced \ifthenelse with \ifdefequal 97
\glsadd: added \glsdetoklabel 177
\glsaddkey: switched to using \@gls@field@link 89
\glsdetoklabel: new 65
\glsdisplaynumberlist: added \glsdetoklabel 176
\glsdoifexistsorwarn: new . . 66
\glsentryaccess: switched to using \@gls@entry@field . 342
\glsentrydescaccess: switched to using \@gls@entry@field 343
\glsentrydescpluralaccess: switched to using \@gls@entry@field 343
\glsentryfirstaccess: switched to using \@gls@entry@field 343
\glsentryfirstplural: added \glsdetoklabel 171
\glsentrylongaccess: switched to using \@gls@entry@field 344
\glsentrylongpluralaccess: switched to using \@gls@entry@field 344
\glsentrypluralaccess: switched to using \@gls@entry@field 343
\glsentryshortaccess: switched to using \@gls@entry@field 343
\glsentryshortpluralaccess: switched to using \@gls@entry@field 343
\glsentrysymbolaccess: switched to using \@gls@entry@field 343
\glsentrysymbolpluralaccess: switched to using \@gls@entry@field 343
\glsentrytextaccess: switched to using \@gls@entry@field 342
\glsgenacfmt: redefined to use accessibility information . . 352
\glsgenentryfmt: redefined to use accessibility information 350
\glshyperlink: added \glsdetoklabel 177
\glslocalreset: added \glsdetoklabel 105
\glslocalunset: added \glsdetoklabel 106
\glsmoveentry: added \glsdetoklabel 101
replaced \ifthenelse with \ifdefequal 101
\glsrefentry: added \glsdetoklabel 231
\glsreset: added \glsdetoklabel 105
\glsseelist: added \expandafter commands . 212
\glsstepentry: added \glsdetoklabel 230
\glsstepsubentry: added \glsdetoklabel 230
\glsunset: added \glsdetoklabel 106
\ifglsdescsuppressed: added \glsdetoklabel 68
fixed typo 68
\ifglsentryexists: added \glsdetoklabel 65
\ifglshaschildren: added \glsdetoklabel 67
\ifglshasdesc: added \glsdetoklabel 67
\ifglshasfield: new 69
\ifglshaslong: added \glsdetoklabel 68
\ifglshasparent: added \glsdetoklabel 67
\ifglshasshort: added \glsdetoklabel 68
\ifglshassymbol: added \glsdetoklabel 68
replaced \ifcsempty with \ifdefempty and replaced \ifx with \ifdefequal 68
\ifglsused: added \glsdetoklabel 65
long-sc-short-desc: redefined to use accessibility information 365
long-short: redefined to use accessibility information . . 364
long-short-desc: redefined to use accessibility information 365

long-sm-short-desc: redefined to use accessibility information 366
\renewacronymstyle: new 249
sc-short-long-desc: redefined to use accessibility information 366
short-long: commented spurious EOL 251
redefined to use accessibility information 364
short-long-desc: redefined to use accessibility information 366
\showglocounter: added \glsdetoklabel 266
\showglodesc: added \glsdetoklabel 267
\showglodescaccess: added \glsdetoklabel 374
\showglodescplural: added \glsdetoklabel 268
\showglodescpluralaccess: added \glsdetoklabel . . . 374
\showglofirst: added \glsdetoklabel 265
\showglofirstaccess: added \glsdetoklabel 374
\showglofirstpl: added \glsdetoklabel 266
\showglofirstpluralaccess: added \glsdetoklabel . . . 374
\showgloflag: added \glsdetoklabel 269
\showgloindex: added \glsdetoklabel 269
\showglolevel: added \glsdetoklabel 265
\showglolong: added \glsdetoklabel 268
\showglolongaccess: added \glsdetoklabel 375
\showglolongpluralaccess: added \glsdetoklabel . . . 375
\showgloname: added \glsdetoklabel 267
\showglonameaccess: added \glsdetoklabel 374
\showgloparent: added \glsdetoklabel 265
\showgloplural: added \glsdetoklabel 265
\showglopluralaccess: added \glsdetoklabel 374
\showgloshort: added \glsdetoklabel 268
\showgloshortaccess: added \glsdetoklabel 375
\showgloshortpluralaccess: added \glsdetoklabel . . . 375
\showglosort: added \glsdetoklabel 268
\showglosymbol: added \glsdetoklabel 268
\showglosymbolaccess: added \glsdetoklabel 374
\showglosymbolplural: added \glsdetoklabel 268
\showglosymbolpluralaccess: added \glsdetoklabel . . . 374
\showglotext: added \glsdetoklabel 265
\showglotextaccess: added \glsdetoklabel 374
\showglotype: added \glsdetoklabel 266
\showglouseri: added \glsdetoklabel 266
\showglouserii: added \glsdetoklabel 266
\showglouseriii: added \glsdetoklabel 267
\showglouseriv: added \glsdetoklabel 267
\showglouserv: added \glsdetoklabel 267
\showglouservi: added \glsdetoklabel 267
sm-short-long-desc: redefined to use accessibility information 367
4.04 – 2014-03-04
\@gls@getcounterprefix: added warning if no prefix can be formed 210
4.04 – 2014-03-06
General: added sort key to printgloss group 229
\@@gls@noidx@nosanitizesort: new 26

\@@gls@noidx@sanitizesort: new 25
\@@gls@nosanitizesort: new . 25
\@@gls@sanitizesort: new . . . 25
\@glo@addchildren: new 218
\@glo@do@sortentries: new . . 219
\@glo@grabfirst: new 224
\@glo@sortedinsert: new 219
\@glo@sortentries: new 218
\@glo@sorthandler@case: new . 220
\@glo@sorthandler@letter: new 220
\@glo@sorthandler@nocase: new 220
\@glo@sorthandler@word: new . 220
\@glo@sortmacro@case: new . . 221
\@glo@sortmacro@def: new . . . 222
\@glo@sortmacro@def@do: new . 222
\@glo@sortmacro@letter: new . 221
\@glo@sortmacro@nocase: new . 222
\@glo@sortmacro@standard: new 221
\@glo@sortmacro@use: new . . . 223
\@glo@sortmacro@word: new . . 221
\@gls@noidx@do: new 224
\@gls@noidx@getgrouptitle: new 235
\@gls@noref@warn: new 200
\@gls@reference: new 227
\@gls@warnonglossdefined: new 24
\@gls@warnontheglossdefined: new 24
\@no@makeglossaries: new . . . 200
\@print@glossary: new 216
\@print@noidx@glossary: new . 223
\@printgloss@setsort: new . . 214
\@printglossary: new 214
alttree: moved hangindent and parindent assignments outside level test 325
\compatibleglossentry: changed \newcommand to \def as is may or may not be defined 337
\compatiblesubglossentry: changed \newcommand to \def as is may or may not be defined 337
\glo@grabfirst: new 224
\gls@defglossaryentry: replaced \ifx with \ifdefvoid 100
\glsnoidxdisplayloc: new . . . 226
\glsnoidxdisplayloclisthandler: new 226
\glsnoidxloclist: new 226
\glsnoidxloclisthandler: new 226
\glsnoidxstripaccents: new . 26
\makeglossaries: Moved definition of \glswrite to \makeglossaries 194
\makenoidxglossaries: new . . 196
\printglossary: changed to use new \@printglossary 213
\printnoidxglossaries: new . 214
\printnoidxglossary: new . . . 214
\showgloloclist: new 269
\warn@noprintglossary: Activate warning in \makeglossaries 213
\writeist: checked for definition of \glswrite 181, 185
4.06 – 2014-03-12
General: added \glsifhyper 158–165
\@GLS@: added \glsifhyper . . . 139
\@GLSpl: added \glsifhyper . . 141
\@Gls@: added \glsifhyper . . . 138
\@Glspl@: added \glsifhyper . 140
\@gls@: added \glsifhyper . . . 137
\@gls@numbersdef: added hook to set toc title 40
\@gls@symbolsdef: added hook to set toc title 40
\@glsdisp: added \glsifhyper 142
\@glspl@: added \glsifhyper . 140
acronym: added hook to set toc title 20
acronyms: added hook to set toc title 20
\glsdefmain: added hook to set toc title 19
4.07 – 2014-04-04
\@glossarysection: added optional argument when using unstarred version 52
\@gls@noidx@do: added \global in case it's used in a tabular-like style 224

\Acrfullplfmt: fixed no case change bug 244
\glsletentryfield: new 166
4.08 – 2014-07-30
General: added entrycounter option to printgloss family 228
added nopostdot option to printgloss family 228
added subentrycounter option to printgloss family 229
explicitly initialise hyper key 120
moved \glsifhyper . . . 158–165
removed \@sACRlongpl 165
removed \@sAcrlongpl 164
removed \@sacrlongpl 164
removed \@sACRlong 163
removed \@sAcrlong 162
removed \@sacrlong 162
removed \@sACRshortpl 161
removed \@sAcrshortpl 160
removed \@sacrshortpl 160
removed \@sACRshort 159
removed \@sAcrshort 158
removed \@sacrshort 158
removed \@sgls@link 122
removed \@sGLSdescplural . 150
removed \@sGlsdescplural . 149
removed \@sglsdescplural . 149
removed \@sGLSdesc 149
removed \@sGlsdesc 148
removed \@sglsdesc 148
removed \@sglsdisp 142
removed \@sGLSfirstplural 147
removed \@sGlsfirstplural 146
removed \@sglsfirstplural 146
removed \@sGLSfirst 145
removed \@sGlsfirst 144
removed \@sglsfirst 144
removed \@sGLSname 148
removed \@sGlsname 147
removed \@sglsname 147
removed \@sGLSplural 146
removed \@sGlsplural 145
removed \@sglsplural 145
removed \@sGLSpl 141
removed \@sGlspl 140
removed \@sglspl 139
removed \@sGLSsymbolplural 152
removed \@sGlssymbolplural 151
removed \@sglssymbolplural 151
removed \@sGLSsymbol 151
removed \@sGlssymbol 150
removed \@sglssymbol 150
removed \@sGLStext 143
removed \@sGlstext 144
removed \@sglstext 143
removed \@sGLSuseriii 155
removed \@sGlsuseriii 154
removed \@sglsuseriii 154
removed \@sGLSuserii 154
removed \@sGlsuserii 153
removed \@sglsuserii 153
removed \@sGLSuseriv 156
removed \@sGlsuseriv 155
removed \@sglsuseriv 155
removed \@sGLSuseri 153
removed \@sGlsuseri 152
removed \@sglsuseri 152
removed \@sGLSuservi 158
removed \@sGlsuservi 157
removed \@sglsuservi 157
removed \@sGLSuserv 157
removed \@sGlsuserv 156
removed \@sglsuserv 156
removed \@sGLS 139
removed \@sGls 138
removed \@sgls 137
removed \@thirdofthree (defined in kernel) 136
removed sPGLS 335
removed sPgls 333
removed spgls 332
removed sPGLSpl 335
removed sPglspl 334
removed spglspl 333
\@ACRlong: added \do@gls@link@checkfirsthyper 357
\@ACRshort: added \do@gls@link@checkfirsthyper 356
\@Acrlong: added \do@gls@link@checkfirsthyper 356
\@Acrshort: added \do@gls@link@checkfirsthyper 355
\@GLS@: moved \glsifhyper . . . 139
moved check for first use to \@gls@link 139

\@GLSpl: moved \glsifhyper . . 141
moved check for first use to \@gls@link 141
\@Gls@: moved \glsifhyper . . . 138
moved check for first use to \@gls@link 138
\@Glspl@: moved \glsifhyper . 140
moved check for first use to \@gls@link 140
\@acrlong: added \do@gls@link@checkfirsthyper 356
\@acrshort: added \do@gls@link@checkfirsthyper 355
\@closegls: new 191
\@gls@: moved \glsifhyper . . . 137
moved check for first use to \@gls@link 137
\@gls@automake: new 191
\@gls@doautomake: new 37
\@gls@field@link: added assignment of \do@gls@link@checkfirsthyper 143
\@gls@forbidtexext: new 71
\@gls@hyp@opt: new 121
\@gls@link: removed redundancy 124
renamed \gls@type to \glstype 124
\@gls@link@checkfirsthyper: new 122
\@glsdisp: moved \glsifhyper 142
moved check for first use to \@gls@link 142
\@glspl@: moved \glsifhyper . 140
moved check for first use to \@gls@link 140
\@ignored@glossaries: new . . 75
\ACRfull: removed \s@ACRfull 245
switched to using \@gls@hyp@opt 245
\Acrfull: removed \@sAcrfull 244
switched to using \@gls@hyp@opt 244
\acrfull: removed \@sacrfull 244
switched to using \@gls@hyp@opt 244
\ACRfullpl: removed \s@ACRfullpl 246
switched to using \@gls@hyp@opt 246
\Acrfullpl: removed \s@Acrfullpl 245
switched to using \@gls@hyp@opt 245
\acrfullpl: removed \s@acrfullpl 245
switched to using \@gls@hyp@opt 245
\ACRlong: switched to using \@gls@hyp@opt 163
\Acrlong: switched to using \@gls@hyp@opt 162
\acrlong: switched to using \@gls@hyp@opt 161
\ACRlongpl: switched to using \@gls@hyp@opt 165
\Acrlongpl: switched to using \@gls@hyp@opt 164
\acrlongpl: switched to using \@gls@hyp@opt 164
\ACRshort: switched to using \@gls@hyp@opt 159
\Acrshort: switched to using \@gls@hyp@opt 158
\acrshort: switched to using \@gls@hyp@opt 158
\ACRshortpl: switched to using \@gls@hyp@opt 161
\Acrshortpl: switched to using \@gls@hyp@opt 160
\acrshortpl: switched to using \@gls@hyp@opt 160
altlongragged4col: fixed bug that displayed description instead of symbol 297
automake: new 36
\forallacronyms: new 63
\GLS: switched to using \@gls@hyp@opt 138
\Gls: switched to using \@gls@hyp@opt 138
\gls: switched to using \@gls@hyp@opt 137
\gls@defglossaryentry: added check for ignored glossary . . 96
\gls@istfilebase: new 48

\glsaddkey: removed
\@sGLS@user@⟨key⟩ 90
removed \@sGls@user@⟨key⟩ 90
removed \@sgls@user@⟨key⟩ 89
switched to using
\@gls@hyp@opt 89, 90
\GLSdesc: switched to using
\@gls@hyp@opt 148
\Glsdesc: switched to using
\@gls@hyp@opt 148
\glsdesc: switched to using
\@gls@hyp@opt 148
\GLSdescplural: switched to
using \@gls@hyp@opt 149
\Glsdescplural: switched to
using \@gls@hyp@opt 149
\glsdescplural: switched to
using \@gls@hyp@opt 149
\glsdisablehyper: added
\KV@glslink@hyperfalse to
definition 136
\glsdisp: switched to using
\@gls@hyp@opt 142
\glsdohyperlink: new 135
\glsdohypertarget: new 134
\glsenablehyper: added
\KV@glslink@hypertrue to
definition 136
\GLSfirst: switched to using
\@gls@hyp@opt 144
\Glsfirst: switched to using
\@gls@hyp@opt 144
\glsfirst: switched to using
\@gls@hyp@opt 144
\GLSfirstplural: switched to
using \@gls@hyp@opt 146
\Glsfirstplural: switched to
using \@gls@hyp@opt 146
\glsfirstplural: switched to
using \@gls@hyp@opt 146
\glsifhyper: deprecated 121
\glslink: switched to using
\@gls@hyp@opt 122
\glslinkcheckfirsthyperhook:
new 123
\glslinkvar: new 121
\GLSname: switched to using
\@gls@hyp@opt 147
\Glsname: switched to using
\@gls@hyp@opt 147
\glsname: switched to using
\@gls@hyp@opt 147
\GLSpl: switched to using
\@gls@hyp@opt 141
\Glspl: switched to using
\@gls@hyp@opt 140
\glspl: switched to using
\@gls@hyp@opt 139
\GLSplural: switched to using
\@gls@hyp@opt 145
\Glsplural: switched to using
\@gls@hyp@opt 145
\glsplural: switched to using
\@gls@hyp@opt 145
\glsspace: new 243
\GLSsymbol: switched to using
\@gls@hyp@opt 150
\Glssymbol: switched to using
\@gls@hyp@opt 150
\glssymbol: switched to using
\@gls@hyp@opt 150
\GLSsymbolplural: switched to
using \@gls@hyp@opt 151
\Glssymbolplural: switched to
using \@gls@hyp@opt 151
\glssymbolplural: switched to
using \@gls@hyp@opt 151
\GLStext: switched to using
\@gls@hyp@opt 143
\Glstext: switched to using
\@gls@hyp@opt 143
\glstext: switched to using
\@gls@hyp@opt 143
\glstreenamefmt: new 318
\GLSuseri: switched to using
\@gls@hyp@opt 152
\Glsuseri: switched to using
\@gls@hyp@opt 152
\glsuseri: switched to using
\@gls@hyp@opt 152
\GLSuserii: switched to using
\@gls@hyp@opt 153
\Glsuserii: switched to using
\@gls@hyp@opt 153
\glsuserii: switched to using
\@gls@hyp@opt 153
\GLSuseriii: switched to using
\@gls@hyp@opt 154
\Glsuseriii: switched to using
\@gls@hyp@opt 154

\glsuseriii: switched to using \@gls@hyp@opt 154
\GLSuseriv: switched to using \@gls@hyp@opt 155
\Glsuseriv: switched to using \@gls@hyp@opt 155
\glsuseriv: switched to using \@gls@hyp@opt 155
\GLSuserv: switched to using \@gls@hyp@opt 156
\Glsuserv: switched to using \@gls@hyp@opt 156
\glsuserv: switched to using \@gls@hyp@opt 156
\GLSuservi: switched to using \@gls@hyp@opt 157
\Glsuservi: switched to using \@gls@hyp@opt 157
\glsuservi: switched to using \@gls@hyp@opt 157
\ifignoredglossary: new 75
\newglossary: added starred version 72
\newignoredglossary: new . . . 74
\ns@newglossary: added \@glotype@⟨*name*⟩@log 72
new 72
\p@gls@hyp@opt: new 121
\PGLS: changed to use \@gls@hyp@opt 335
\Pgls: changed to use \@gls@hyp@opt 333
\pgls: changed to use \@gls@hyp@opt 332
\PGLSpl: changed to use \@gls@hyp@opt 335
\Pglspl: changed to use \@gls@hyp@opt 334
\pglspl: changed to use \@gls@hyp@opt 333
\s@gls@hyp@opt: new 121
\s@newglossary: new 72
4.09 – 2014-08-12
\glsaddkey: fixed bug in user commands 89
4.10 – 2014-08-27
\@Gls@acrentryname: new 168
\@Gls@entryname: new 167
\@gls@glossary: Renamed \@glossary to \@gls@glossary 202
alttree: moved space after symbol 325, 326
\glspercentchar: new 179
\glstildechar: new 179
4.11 – 2014-09-01
\@@do@esc@wrglossary: added hook 208
\gls@wrglossary: renamed from \@wrglossary to \gls@wrglossary 203
\glsaddprotectedpagefmt: new 204
\glsbackslash: new 179
sanitize: none option 29
4.12 – 2014-11-22
\@gls@addpredefinedattributes: Added glsignore attribute . . 57
\@gls@adjustmode: new 178
\@gls@notranslatorhook: removed 29
\@gls@toc: added \protect to \numberline 54
\@gls@usetranslator: new . . . 29
\glsacrpluralsuffix: new . . . 45
\glsadd: added check for vertical mode 177
\glsaddallunused: replaced @gobble with glsignore . . . 178
\glsifusedtranslatordict: new 29
\glsignore: new 178
\glsupacrpluralsuffix: new . 45
\ProvidesGlossariesLang: new 45
\RequireGlossariesLang: new . 45
4.13 – 2015-02-03
\indexspace: new . . . 278, 299, 318
4.14 – 2015-02-28
\@@glslocalreset: new 107
\@@glslocalunset: new 106
\@@glsreset: new 107
\@@glsunset: new 106
\@@newglossaryentry@defcounters: new 108
\@cGls: new 112
\@cGls@: new 112
\@cGlspl@: new 113
\@cgls: new 111
\@cgls@: new 111
\@cglspl: new 112, 113

\@cglspl@: new 112
\@gls@entry@count: new 111
\@gls@increment@currcount: new 110
\@gls@local@increment@currcount: new 110
\@gls@write@entrycounts: new 110
\@glslocalreset: new 106
\@glslocalunset: new 106
\@glsreset: new 107
\@glsunset: new 106
\@newglossaryentry@defcounters: new 101
\cGls: new 111
\cgls: new 111
\cGlsformat: new 112
\cglsformat: new 111
\cGlspl: new 112
\cglspl: new 112
\cGlsplformat: new 113
\cglsplformat: new 112
\gls@defdocnewglossaryentry: new 83
\glsenableentrycount: new . . 108
\glslocalreset: switched to \@glslocalreset 105
\glslocalunset: switched to \@glslocalunset 106
\glsreset: switched to \@glsreset 105
\glsunset: switched to \@glsunset 106
4.15 – 2015-03-16
General: bug fix replaced \@glo@type with \glstype 165
4.16 – 2015-06-18
\glsaddstoragekey: new 87
4.16 – 2015-07-08
General: added \glspostlinkhook . . . 158–165
\@ACRlong: added \glspostlinkhook 357
\@ACRshort: added \glspostlinkhook 356
\@Acrlong: added \glspostlinkhook 357
\@Acrshort: added \glspostlinkhook 356
\@GLS@: added \glspostlinkhook 139
\@GLSpl: added \glspostlinkhook 141
\@Gls@: added \glspostlinkhook 138
\@Glspl@: added \glspostlinkhook 141
\@acrlong: added \glspostlinkhook 356
\@acrshort: added \glspostlinkhook 355
\@gls@: added \glspostlinkhook 138
\@gls@@link: added \glspostlinkhook 122
\@gls@field@link: added \glspostlinkhook 143
\@gls@link: moved definition of \glsifhyperon outside of this macro 124
\@glsdisp: added \glspostlinkhook 142
\@glspl@: added \glspostlinkhook 140
\glsacspace: new 251
\glsadd: changed \@do@wrglossary to \@@do@wrglossary 177
\glsfielddef: new 92
\glsfieldedef: new 91
\glsfieldfetch: new 92
\glsfieldgdef: new 91
\glsfieldxdef: new 91
\glsifhyperon: moved definition of \glsifhyperon 123
\glslinkpostsetkeys: new . . . 123
\glspostlinkhook: new 122
\glswriteentry: new 203
\ifglsfieldcseq: new 94
\ifglsfielddefeq: new 93
\ifglsfieldeq: new 93
long-sp-short: new 250
long-sp-short-desc: new 253
\showglofield: new 269
4.18 – 2015-09-09
General: split mfirstuc into separate bundle 4
4.19 – 2015-10-31
\glstreenamebox: new 324

4.19 – 2015-11-22
\@gls@link@nocheckfirsthyper: new . . . 142
\@gls@preglossaryhook: new . 214
\@printglossary: added \@gls@preglossaryhook . . 216
\do@glsdisablehyperinlist: new . . . 123
\doifglossarynoexistsordo: new . . . 66
\gls@gobbleopt: new . . . 72
\glsdoifexistsordo: new . . . 66
4.20 – 2015-11-30
\@gls@link: added \@gls@setdefault@glslink@opts . . . 124
added \glsdonohyperlink when hyperlink is suppressed 124
\@gls@setdefault@glslink@opts: new . . . 123
\gls@checkseeallowed@preambleonly: new . . . 78
\glsdonohyperlink: new . . . 136
4.21 – 2016-01-24
General: changed checkfirsthyper assignment . . . 158–165
\@printglossary: warn if no style has been set . . . 214
altlong4col-booktabs: new . . 290
altlongragged4col-booktabs: new . . . 291
\glsLTpenaltycheck: new . . . 292
\glspatchLToutput: new . . . 292
\glspenaltygroupskip: new . . 292
long-booktabs: new . . . 288
long3col-booktabs: new . . . 289
long4col-booktabs: new . . . 289
longragged-booktabs: new . . . 290
longragged3col-booktabs: new 291
\setglossarystyle: set default style if not already set . . . 237
4.22 – 2016-04-19
\@@do@esc@wrglossary: added check for \@arabic . . . 207
added test to allow temporary primitive modifications and added arabic case . . . 207
\gls@arabicpage: new . . . 204
\gls@protected@pagefmts: added arabic to list . . . 203
\glsentrytitlecase: new . . . 171
\glsfindwidesttoplevelname: new . . . 324
\glslistgroupheaderfmt: new . 278
\glslistnavigationitem: new . 278
\glstreegroupheaderfmt: new . 318
\glstreenavigationfmt: new . 318
\ifglswrallowprimitivemods: new . . . 206
list: fixed missing space before description . . . 279
long: fixed typo in \glossentrydesc . . . 282
mcolalttreespannav: new . . . 304
mcolindexspannav: new . . . 300
mcoltreenonamespannav: new . 302
mcoltreespannav: new . . . 301
super4col: fixed bug in \glossentry . . . 309
4.23 – 2016-04-30
altlongragged4col: check for nogroupskip changed . . . 297
altsuperragged4col: check for nogroupskip changed . . . 316
\glscurrentfieldvalue: new . 70
\ifglshasfield: added \glscurrentfieldvalue 69, 70
long: check for nogroupskip changed . . . 283
long-booktabs: check for nogroupskip changed . . . 289
long3col: check for nogroupskip changed . . . 284
long3col-booktabs: check for nogroupskip changed . . . 289
long4col: check for nogroupskip changed . . . 286
long4col-booktabs: check for nogroupskip changed . . . 290
longragged: check for nogroupskip changed . . . 294
longragged3col: check for nogroupskip changed . . . 295
super: check for nogroupskip changed . . . 305
super3col: check for nogroupskip changed . . . 307
super4col: check for nogroupskip changed . . . 309

superragged: check for nogroupskip changed 313
superragged3col: check for nogroupskip changed 315
4.24 – 2016-05-27
\@gls@extramakeindexopts: new 189
\@gls@glossary: added check for debug mode 202
\@gls@see@noindex: new 9
debug: new 7
\glsnomakeindexwarning: new . 54
\GlsSetQuote: new 186
\GlsSetWriteIstHook: new . . . 186
seenoindex: new 9
4.25 – 2016-06-09
General: changed \DeclareRobustCommand to \newrobustcmd and changed \@ifundefined to \ifcsundef 349
\@gls@enablesavenonumberlist: new 79
\@gls@initnonumberlist: new 78, 79
\@gls@savenonumberlist: new . 78
4.26 – 2016-10-12
\@glossary@default@style: added check for classicthesis 11
\glstreechildpredesc: new . . 319
\glstreeitem: new 319
\glstreepredesc: new 319
\glstreesubitem: new 319
\glstreesubsubitem: new 319
mcolindex: replaced \@idxitem with \glstreeitem 299
mcolindexspannav: replaced \@idxitem with \glstreeitem 300
4.28 – 2017-01-07
\glspatchtabularx: new 105
4.29 – 2017-01-19
General: added check for \@glsxtr@doaccsupp 337
\@gls@noidx@do: current letter group assignment made global 225
\@print@noidx@glossary: moved definition of \@gls@currentlettergroup outside of theglossary environment 223
\glsnavhyperlinkname: new . . 272
4.30 – 2017-06-11
\@glo@autosee: new 100
\@glo@autoseehook: new 101
\@glo@check@sortallowed: new 16
\@gls@noidx@do: letter group assignment made global . . . 225
\@gls@setupsort@def: added check for register 17
\@gls@setupsort@none: new . . 18
\@xdycrossrefhook: new 59
\@xdylocationclassorder: bug fix: changed \edef to \def . 60
\glosortentrieswarning: new . 23
\gls@set@xr@key: new 77
\gls@xr@key: new 77
\GlsAddXdyLocation: bug fix: changed #1 to #2 60
\glsnoidxstripaccents: added \a 26
added \TH, \dh and \DH 27
4.31 – 2017-08-10
nolist: added check for "list" style 13
4.31 – 2017-09-10
style: changed \renewcommand to \def 11
4.32 – 2017-08-24
\@glsnavhypertarget: new . . . 273
\@glsshowtarget: new 8
\glsshowtarget: new 8
4.33 – 2017-09-20
\@@do@esc@wrglossary: added \gls@the and \gls@number 208
renamed from \@@do@esc@wrglossary . . . 206
\@@do@noesc@wrglossary: new . 205
\@@do@wrglossary: changed to check for esclocations 205
\@gls@missinglang@warn: new . 23
esclocations: new 12
\GlsSetXdyFirstLetterAfterDigits: added starred version 179
\GlsSetXdyNumberGroupOrder: new 180

4.34 – 2017-11-03
\glsshowtarget: modified to check for math mode and inner 8
mcolalttreespannav: removed spurious space 304
4.35 – 2017-11-14
\glsadd: added \@gls@setsort (in case of sort=use) 177
4.36 – 2018-03-07
\@gls@glossary: removed \index 202
4.37 – 2018-04-07
\gls@begindocdefs: new 85
4.38 – 2018-05-10
\@gls@define@glossaryentrycounter: added check for existence of glossaryentry counter 14
new 14
\@gls@define@glossarysubentrycounter: new 15
prepended \currentglossary. to \theHglossarysubentry and removed spurious eol space 15
\glsaccsupp: added braces around actual text argument 346
\glsentrycounterlabel: bug fix: move conditional inside command 231
\GlsEntryCounterLabelPrefix: new 228
\glsentryitem: bug fix: move conditional inside command 231
\glsrefentry: bug fix: move conditional inside command 231
\glsreseteentrycounter: bug fix: move conditional inside command 230
\glsresetsubentrycounter: bug fix: move conditional inside command 230
\glsstepentry: bug fix: move conditional inside command 230
\glsstepsubentry: bug fix: move conditional inside command 230
\glssubentrycounterlabel: bug fix: move conditional inside command 231
\glssubentryitem: bug fix: move conditional inside command 231
\showglonameaccess: bug fix: corrected field (was showing text access field) 374
4.40 – 2018-06-01
\istfile: changed \def to \providecommand 200
\makenoidxglossaries: false . . 197
4.41 – 2018-07-23
General: changed \val and \nr to \gls@numberedsection@val and \gls@numberedsection@nr 228
\@gls@override@glossary: new 39
debug: changed \val and \nr to \gls@debug@val and \gls@debug@nr 7
\glossary: added warning 39
\gls@original@glossary: new . 38
\gls@original@makeglossary: new 38
kernelglossredefs: new 39
\makeglossaries: removed redefinition of \makeglossary 195
\makeglossary: added warning 38
nonumberlist: changed \val and \nr to \gls@nonumberlist@val and \gls@nonumberlist@nr 78
numberedsection: changed \val and \nr to \gls@numberedsection@val and \gls@numberedsection@nr . 10
seenoindex: changed \val and \nr to \gls@seenoindex@val and \gls@seenoindex@nr . . . 9
translate: changed \val and \nr to \gls@translate@val and \gls@translate@nr 30
4.42 – 2019-01-06
\@gls@@automake@immediate: new 193
\@gls@automake@immediate: new 192
altlong4col-booktabs: removed superfluous \glspatchLToutput 290

automake: switch from boolean to choice 36
\gls@automake@nr: new 36
\glsfieldedef: changed from \edef to \protected@csedef 91
\glsfieldxdef: changed from \edef to \protected@csxdef 91
\ifglsautomake: now defined explicitly instead of through boolean key 36
noglossaryindex: new 41
4.43 – 2019-09-28
\glsnoidxstripaccents: add check for LaTeX version 2019/10/01 27
4.44 – 2019-12-06
\@glsprefix@record@hook: new 332
4.45 – 2020-02-13
General: removed backward compatibility use of symbol key 340
\@@do@write@glslabels: new . 35
\@@glsshowtarget: new 8
\@GLSdesc@: added accessibility support 359
\@GLSdescplural@: added accessibility support 359
\@GLSfirst@: added accessibility support 358
\@GLSfirstplural@: added accessibility support 358
\@GLSname@: added accessibility support 359
\@GLSplural@: added accessibility support 358
\@GLSsymbol@: added accessibility support 360
\@GLSsymbolplural@: added accessibility support 360
\@GLStext@: added accessibility support 357
\@GLSuseri@: added accessibility support 360
\@GLSuserii@: added accessibility support 361
\@GLSuseriii@: added accessibility support 361
\@GLSuseriv@: added accessibility support 361
\@GLSuserv@: added accessibility support 362
\@GLSuservi@: added accessibility support 362
\@Glsdesc@: added accessibility support 359
\@Glsdescplural@: added accessibility support 359
\@Glsfirst@: added accessibility support 358
\@Glsfirstplural@: added accessibility support 358
\@Glsname@: added accessibility support 359
\@Glsplural@: added accessibility support 358
\@Glssymbol@: added accessibility support 360
\@Glssymbolplural@: added accessibility support 360
\@Glstext@: added accessibility support 357
\@Glsuseri@: added accessibility support 360
\@Glsuserii@: added accessibility support 361
\@Glsuseriii@: added accessibility support 361
\@Glsuseriv@: added accessibility support 361
\@Glsuserv@: added accessibility support 362
\@Glsuservi@: added accessibility support 362
\@do@write@glslabels: new . . 35
\@domakeglossaries: new 34
\@gls@access@display: deprecated 346
\@gls@fieldaccess@display: new 346
\@glsdesc@: added accessibility support 359
\@glsdescplural@: added accessibility support 359
\@glsfirst@: added accessibility support 358
\@glsfirstplural@: added accessibility support 358
\@glsname@: added accessibility support 359

\@glsplural@: added accessibility support 358
\@glsshowaccsupp: new 9
\@glssymbol@: added accessibility support 360
\@glssymbolplural@: added accessibility support 360
\@glstext@: added accessibility support 357
\@glsuseri@: added accessibility support 360
\@glsuserii@: added accessibility support 361
\@glsuseriii@: added accessibility support 361
\@glsuseriv@: added accessibility support 361
\@glsuserv@: added accessibility support 362
\@glsuservi@: added accessibility support 362
\@org@Gls@acrentryname: added check for \glsshortaccessdisplay . 168
acronyms: changed \renewcommand to \def 20
debug: showaccsupp 7
disablemakegloss: new 34
\gls@accessibility: new 337
\gls@accsupp@engine: new . . . 337
\glsaccessibility: new 337
\glsdefaultshortaccess: new . 373
\glsdescriptionaccessdisplay: added check for existence . 347
\glsdescriptionpluralaccessdisplay: added check for existence . 348
\glsentrydescpluralaccess: corrected field reference . . . 343
\glsentryfirstpluralaccess: switched to using \@gls@entry@field 343
\glsentryparent: new 172
\Glsentryprefix: added \glsdetoklabel 331
\glsentryprefix: added \glsdetoklabel 330
\Glsentryprefixfirst: added \glsdetoklabel 330
\glsentryprefixfirst: added \glsdetoklabel 330
\Glsentryprefixfirstplural: added \glsdetoklabel . . . 331
\glsentryprefixfirstplural: added \glsdetoklabel . . . 330
\Glsentryprefixplural: added \glsdetoklabel 331
\glsentryprefixplural: added \glsdetoklabel 330
\glsentrytitlecase: added existence check 171
\glsentryuseriaccess: new . . 344
\glsentryuseriiaccess: new . 344
\glsentryuseriiiaccess: new . 344
\glsentryuserivaccess: new . 344
\glsentryuservaccess: new . . 344
\glsentryuserviaccess: new . 344
\glsfieldaccsupp: new 345
\glsfirstaccessdisplay: added check for existence 347
\glsfirstpluralaccessdisplay: added check for existence . 347
\glslongaccessdisplay: added check for existence 348
\glslongpluralaccessdisplay: added check for existence . 348
\glsnameaccessdisplay: added check for existence 346
\glspluralaccessdisplay: added check for existence . 347
\glsprefixsep: new 332
\glssee: switched to \newrobustcmd 211
\glsseeformat: switched to \newrobustcmd 211
\glsseeitem: switched to \newrobustcmd 212
\glsseelist: switched to \newrobustcmd 211
\glsshortaccessdisplay: added check for existence 348
\glsshortaccsupp: new 346
\glsshortplaccsupp: new 346
\glsshortpluralaccessdisplay: added check for existence . 348
\glsshowaccsupp: new 9
\glsshowtargetfont: new 8
\glsshowtargetouter: new 8
\glsshowtargetsymbol: new . . . 8
\glssymbolaccessdisplay: added check for existence . 347

\glssymbolpluralaccessdisplay: added check for existence . 347
\glstextaccessdisplay: added check for existence 347
\glsuseriaccessdisplay: new . 348
\glsuseriiaccessdisplay: new 348
\glsuseriiiaccessdisplay: new 349
\glsuserivaccessdisplay: new 349
\glsuservaccessdisplay: new . 349
\glsuserviaccessdisplay: new 349
\ifglshaschildren: made robust 67
\ifglshasfield: made robust . 69
\ifglshaslong: made robust . . 68
\ifglshasprefix: added \glsdetoklabel 331
\ifglshasprefixfirst: added \glsdetoklabel 331
\ifglshasprefixfirstplural: added \glsdetoklabel . . . 332
\ifglshasprefixplural: added \glsdetoklabel 331
\ifglshasshort: made robust . 68
\ifglshassymbol: made robust 68
\makeglossaries: let \@makeglossary to \@gobble instead of \relax 195
restoremakegloss: new 34
user1access: new 339
user2access: new 339
user3access: new 339
user4access: new 340
user5access: new 340
user6access: new 340
writeglslabels: new 36
\xglsfieldaccsupp: new 345

4.46 – 2020-03-19

\@ifglossaryexists: new 64
\@printgloss@checkexists: new 213
\@printgloss@checkexists@allowignored: new 213
\@printgloss@checkexists@noignored: new 213
\@printglossary: replaced \ifglossaryexists with \@printgloss@checkexists 215
\doifglossarynoexistsordo: switched to starred form of \ifglossaryexists 66
\ifglossaryexists: added starred form 64
\s@ifglossaryexists: new . . . 64
\setglossarypreamble: switched to starred form of \ifglossaryexists 50

4.47 – 2021-09-20

General: changed \x to \@gls@x 239
\@@do@esc@wrglossary: changed \edef to \protected@edef 208
\@@do@noesc@wrglossary: changed \edef to \protected@edef 206
\@@do@write@glslabels: fully expand list 35
\@@gls@@default@acr@descplural: new 261
\@@gls@@default@acr@firstplural: new 261
\@@gls@@default@acr@plural: new 261
\@@gls@noexpand@assign@default@plurals: new 261
\@glo@do@sortentries: changed \edef to \protected@edef 219
\@glo@sortentries: changed \edef to \protected@edef 218
\@glo@storeentry: changed \edef to \protected@edef 102
\@gls@checkesc: new 132
\@gls@deprecated@acr@warning: new 263
\@gls@link: changed \edef to \protected@edef 123
\@gls@setupsort@use: changed \edef to \protected@edef . 18
\@gls@writedef: added a warning if entry has already been defined 86
\@makeglossaries@warn@noprintglossary: new 196
\@write@glslabelsname: new . 35
\do@write@glslabelsname: new 35
\forglsentries: changed \edef to \protected@edef 64
\gls@defdocnewglossaryentry: removed see warning 83
\gls@defglossaryentry: changed \edef to \protected@edef 95, 96

\gls@docdef@addlabel: new . . 84
\gls@docdef@labellist: new . 84
\gls@glsdefs@addlabel: new . 84
\gls@glsdefs@labellist: new . 84
\gls@save@numberlist: changed \edef to \protected@edef 212
\glsadd: changed \edef to \protected@edef 177
\glsaddall: changed \edef to \protected@edef 178
\glsdefs@newdocentry: new . . 85
\glsdisplaynumberlist: changed \edef to \protected@edef 176
\glsfielddef: changed \edef to \protected@edef 92
\glsfieldedef: changed \edef to \protected@edef 91
\glsfieldfetch: changed \edef to \protected@edef 92
\glsfieldgdef: changed \edef to \protected@edef 91
\glsfieldxdef: changed \edef to \protected@edef 91
\glsIfListOfAcronyms: changed \edef to \protected@edef . 21
\glsmoveentry: changed \edef to \protected@edef 101
\glsnavhyperlink: changed \edef to \protected@edef 271
\GlsSetQuote: added check for backslash 188
\glsstepsubentry: changed \edef to \protected@edef 231
\ifglsfieldcseq: changed \edef to \protected@edef 94
\ifglshaschildren: changed \edef to \protected@edef . 67
\ifignoredglossary: changed \edef to \protected@edef . 75
\makenoidxglossaries: changed \edef to \protected@edef 197
\new@glossaryentry: allow overwrite if defined in .glsdefs 84
\newignoredglossary: changed \edef to \protected@edef . 74
\ns@newglossary: changed \edef to \protected@edef 72
\SetDescriptionDUAAcronymStyle: added deprecated warning . 262
\SetDescriptionFootnoteAcronymStyle: added deprecated warnings 262
\SetDUAStyle: added deprecated warnings 263
\SetFootnoteAcronymStyle: added deprecated warnings 263
\SetSmallAcronymStyle: added deprecated warnings 263
writeglslabelnames: new 36
4.48 – 2021-10-19
\glscapitalisewords: new . . . 171
\glslistexpandedname: new . . 278
\glslistinit: new 278
\glsunexpandedfieldvalue: new 92
4.49 – 2021-11-01
\glslistinit: added \glsnavigation 278
4.50 – 2022-10-14
General: added initialisation . . . 242
allow markup to be stripped from location 239
corrected description mapping 340
don't load glossary-super if xtab has been loaded 12
No longer requires textcase with mfirstuc 2.08+ 5
removed automatic loading of glossaries-compatible-307 . . . 42
switched to using \glsuppercase 143, 145–159, 161, 163, 165
\@@do@esc@wrglossary: moved \romannumeral inside allow primitive mods 208
\@@gls@@default@acr@descplural: removed 261
\@@gls@@default@acr@firstplural: removed 261
\@@gls@@default@acr@plural: removed 261
\@@gls@default@entryfmt: removed 349
\@@gls@navhypertarget: new . 273
\@@gls@noexpand@assign@default@plurals: removed 261
\@Gls@acrentryname: removed mfirstuc internal commands 168
\@Glsdisp: new 142
\@Glsentryfield: new 167

\@domakeglossaries: added check for \nofiles 34
\@gls@automake: added check for void codepage 191
\@gls@automake@immediate: added check for void codepage 192
\@gls@automake@types: new . . 193
\@gls@defaulttype: new 71
\@gls@do@automake@err: new . 36
\@gls@do@glsunset: new 120
\@gls@entry@field: switched to \csuse 166
\@gls@escbsdq: check esclocations setting 127
\@gls@getcounterprefix: changed \edef to \def . . . 210
\@gls@getgrouptitle: switched to LaTeX3 syntax 235
\@gls@glslink@hyper@update@hook: new 120
\@gls@hypergroup: updated to use etoolbox 274
\@gls@noist@toolate: new . . . 190
\@gls@removedoption: new . . . 27
\@gls@requested@languages: new 31
\@gls@restore@glslocal: new . 120
\@gls@sanitizedesc: removed . 24
\@gls@save@glslocal: new . . . 120
\@gls@setupsort@clear: new . 18
\@glsdefaultplural: Removed 80
\@org@Gls@acrentryname: saved original definition of \@Gls@acrentryname 168
\@xp@gls@getcounterprefix: new 209
\@xp@glshypernumber: new . . . 238
automake: added makegloss and lite options 36
automakegloss: new 37
automakeglosslite: new 37
compatible-2.07: deprecated . . 38
compatible-3.07: deprecated . . 38
\compatibleglossentry: removed 232, 337
\compatiblesubglossentry: removed 234, 337
\CustomAcronymFields: removed 264
\CustomNewAcronymDef: removed 264
\DefaultNewAcronymDef: removed 261, 373
\defglsdisplay: removed 119
\defglsdisplayfirst: removed 119
\DescriptionDUANewAcronymDef: removed 262
\DescriptionFootnoteNewAcronymDef: removed 262, 373
\DescriptionNewAcronymDef: removed 262, 373
dua: switched to using \glsuppercase 255, 256
dua-desc: removed \acronymfont from around long form . . . 256
\DUANewAcronymDef: removed . . 263
esclocations: default now false 12
\findrootlanguage: removed . . 62
footnote: switched to using \glsuppercase 257
\FootnoteNewAcronymDef: removed 263, 373
\glossary: removed 39
\glossaryentryfield: removed 234
\glossarymark: switched to using \glsuppercase 52
\glossarystyle: removed 237
\glossarysubentryfield: removed 234
\gls@automake@makegloss: new 36
\gls@automake@option: new . . 36
\gls@codepage: set default codepage to utf8 if \inputencodingname not defined and switched to \ifdefvoid 33
\gls@delimN: renamed \@delimN to \gls@delimN 239
\gls@delimR: renamed \@delimR to \gls@delimR 239
\gls@disablepagerefexpansion: added check for esclocations 203
\gls@noidx@nonumberlist: new 78
\gls@noidxglossary: added \glswrglossdisablelocationcmds 200
\gls@xdy@set@lang: new 33
\glsaddkey: allow expansion in PDF bookmarks 89

switched to using \glsuppercase 90
\GlsAddXdyAttribute: remove duplicate counterless attribute definitions 56
\Glsdisp: new 142
\glsdisplay: removed 119
\glsdisplayfirst: removed . . . 119
\glsdoparenifnotempty: replaced \DeclareRobustCommand with \newrobustcmd 263
\glsdoshowtarget: new 8
\glsencapwrcontent: new 124
\Glsentrydesc: allow expansion in PDF bookmarks 169
\Glsentrydescplural: allow expansion in PDF bookmarks 169
\Glsentryfirst: allow expansion in PDF bookmarks 171
\Glsentryfirstplural: allow expansion in PDF bookmarks 171
\glsentryfmt: obsolete commands replaced with \glsgenentryfmt 114
\Glsentrylong: allow expansion in PDF bookmarks 174
\Glsentrylongpl: allow expansion in PDF bookmarks 174
\Glsentryname: allow expansion in PDF bookmarks 167
\Glsentryplural: allow expansion in PDF bookmarks 170
\Glsentryprefix: allow expansion in PDF bookmarks 331
\Glsentryprefixfirst: allow expansion in PDF bookmarks 330
\Glsentryprefixfirstplural: allow expansion in PDF bookmarks 331
\Glsentryprefixplural: allow expansion in PDF bookmarks 331
\Glsentryshort: allow expansion in PDF bookmarks 174
\Glsentryshortpl: allow expansion in PDF bookmarks 174
\Glsentrysymbol: allow expansion in PDF bookmarks 170
\Glsentrysymbolplural: allow expansion in PDF bookmarks 170
\Glsentrytext: allow expansion in PDF bookmarks 170
\glsentrytitlecase: use sentence case for PDF bookmark 171
\Glsentryuseri: allow expansion in PDF bookmarks 172
\Glsentryuserii: allow expansion in PDF bookmarks 172
\Glsentryuseriii: allow expansion in PDF bookmarks 173
\Glsentryuseriv: allow expansion in PDF bookmarks 173
\Glsentryuserv: allow expansion in PDF bookmarks 173
\Glsentryuservi: allow expansion in PDF bookmarks 174
\glsfielddef: made robust . . . 92
\glsfieldedef: made robust . . 91
\glsfieldfetch: made robust . 92
\glsfieldgdef: made robust . . 91
\glsfieldxdef: made robust . . 91
\glsgenacfmt: switched to using \glsuppercase 117
\glsgenentryfmt: switched to using \glsuppercase 115
\glsgetgrouplabel: converted to use LaTeX3 236
\glsgetgrouptitle: made robust 235
\glsglossarymark: switched to \glsuppercase 51
\glsifhyper: removed 121
\glsindexingsetting: new . . . 15
\glsinlineifhaschildren: new 276

\Glslink: new 122
\glslocationcstoencap: new . 50
\glslowercase: new 4
\glsmfuaddmap: new 5
\glsmfublocker: new 5
\glsmfuexcl: new 5
\glsnavhyperlink: changed \@glo@type to \currentglossary 271
\glsnavhypertarget: changed \@glo@type to \currentglossary 273
\glsnavigation: changed \@glo@type to \currentglossary 274
\glsnoidxprenumberlist: new . 78
\glsnumberlistloop: corrected parameter number 200
\glsseeitemformat: added test for short field 212
\glssentencecase: new 4
\glsshowtargetfonttext: new . . 8
\glsshowtargetinner: new 8
\glstexorpdfstring: new 167
\glsuppercase: new 4
\GlsWarnAddProtectedPageFmt: new 205
\glswrglossdisableanchorcmds: new 205
\glswrglossdisablelocationcmds: new 204
\glswrglosslocationtarget: new 205
\glswrglosslocationtextfmt: new 205
\ifglsfieldcseq: made robust 94
\ifglsfielddefeq: made robust 93
\ifglsfieldeq: made robust . . 93
\ifglsfieldvoid: new 68
\ifglsresetcurrcount: new . . 108
\ifglswrallowprimitivemods: switch default to false 206
\istfile: removed 200
kernelglossredefs: corrected names of choice variables . . . 39
languages: new 31
\makeglossary: removed 38
mfirstuc: new 41
\SetAcronymStyle: removed . . . 264
\SetCustomDisplayStyle: removed 264
\SetCustomStyle: removed . . . 264
\SetDefaultAcronymStyle: removed 261
\SetDescriptionAcronymDisplayStyle: removed 262
\SetDescriptionAcronymStyle: removed 262
\SetDescriptionDUAAcronymDisplayStyle: removed 262
\SetDescriptionDUAAcronymStyle: removed 262
\SetDescriptionFootnoteAcronymDisplayStyle: removed 262
\SetDescriptionFootnoteAcronymStyle: removed 262
\SetDUADisplayStyle: removed 263
\SetDUAStyle: removed 263
\SetFootnoteAcronymDisplayStyle: removed 262
\SetFootnoteAcronymStyle: removed 263
\SetGenericNewAcronym: corrected long plural suffix 247
switched to using \glsuppercase 247
\setglossentrycompatibility: removed 234
\SetSmallAcronymDisplayStyle: removed 263
\SetSmallAcronymStyle: removed 263
\SmallNewAcronymDef: removed 263, 373
sort: added clear 15

4.51 – 2022-11-02

General: switch to \TrackIfKnownLanguage . . . 45
\@do@wrglossary: added \gls@ifnotmeasuring 203
\@gls@@automake@immediate: ensure list is expanded before iterating 193
\@gls@if@measuring: new 103
alttree: replaced \settowidth with \glsmeasurewidth 325, 326
\gls@if@measuring: new 103
\gls@local@stepcounter: new . 104
\gls@org@refstepcounter: new 103

\gls@start@measuring: new . . 104
\gls@stop@measuring: new . . . 104
\glsdohypertarget: added measuring and group 134
\glsfindwidesttoplevelname: replaced \settowidth with \glsmeasurewidth 324
\glsifmeasuring: new 104
\glsmeasuredepth: new 105
\glsmeasureheight: new 105
\glsmeasurewidth: new 105
\glspatchtabularx: switch measuring on and off 105
\makeglossaries: ensure list of glossaries is expanded before iterating 196
4.52 – 2022-11-03
\gls@local@stepcounter: corrected spelling of \@ne . 104
4.53 – 2023-09-29
\@gls@hypergroup: rewritten in LaTeX3 274
\glsnavhypergroupdotarget: new 273
\glsnavigationitem: new 274
4.54 – 2024-04-03
General: added check for \l_datatool_measure_hook_tl 104
\gls@printglossary@begin: new 216
\gls@printglossary@end: new . 216
\glsdohyperlinkhook: new . . . 135
\glsdohypertargethook: new . 134
\Glsentrydesc: change map to exclusion 169
\Glsentrydescplural: change map to exclusion 169
\Glsentryfirst: change map to exclusion 171
\Glsentryfirstplural: change map to exclusion 171
\Glsentryfull: change map to exclusion 175
\Glsentrylong: change map to exclusion 174
\Glsentrylongpl: change map to exclusion 174
\Glsentryname: change map to exclusion 167
\Glsentryplural: change map to exclusion 170
\Glsentryshort: change map to exclusion 174
\Glsentryshortpl: change map to exclusion 174
\Glsentrysymbol: change map to exclusion 170
\Glsentrysymbolplural: change map to exclusion 170
\Glsentrytext: change map to exclusion 170
\Glsentryuseri: change map to exclusion 172
\Glsentryuserii: change map to exclusion 172
\Glsentryuseriii: change map to exclusion 173
\Glsentryuseriv: change map to exclusion 173
\Glsentryuserv: change map to exclusion 173
\Glsentryuservi: change map to exclusion 174
\glslabelhypertarget: new . . 134
\glslabelhypertargetdefs: new 135
\glslabelhypertargetprefix: new 135
\glslabelhypertargetvalue: new 135
\glslowercase: ensure argument is expanded 4
\glsuppercase: ensure argument is expanded 4

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

\! . . . 128, 130, 131, 188, 450, 451, 699, 701, 702, 732, 953, 955, 986
\" 26, 127, 129–131, 133, 387, 448–450, 452, 637, 699–702, 704, 891, 952–955, 957, 958
\# 132, 133, 183, 475, 476, 703, 704, 727, 957, 982
\% 179, 185, 473, 477, 578–580, 725, 729, 831–833, 979, 984
\& 45, 176, 398, 472, 648, 724, 904, 978
\' 26, 387, 637, 891
\. . 13, 26, 380, 387, 630, 637, 884, 891
\= 26, 387, 637, 891
\? . . . 127, 129, 130, 188, 448–450, 479, 699–701, 731, 952, 954, 986
\@ 85, 420, 671, 927
\@@delimN 510, 762
\@@do@@wrglossary 197, 206, 208, 209, 485, 491, 492, 737, 742, 743, 992, 998, 999
\@@do@esc@wrglossary 205, 206, 490, 491, 742, 743, 998
\@@do@noesc@wrglossary 205, 205, 490, 742, 998
\@@do@wrglossary 177, 203, 205, 472, 489, 490, 724, 741, 742, 979, 996, 998
\@@do@write@glslabels 35, 35, 393, 643, 898
\@@glo@assign@sortkey 198, 229, 486, 504, 738, 756, 993, 1012
\@@glo@list 64, 409, 659, 915
\@@glo@sort 26, 387, 637, 891
\@@glo@type 214, 494, 746, 1002
\@@glossarysec . . . 10, 10, 52, 53, 379, 402, 628, 629, 652, 883, 908
\@@glossaryseclabel 10, 10, 11, 52, 53, 228, 379, 402, 504, 629, 652, 755, 756, 883, 908, 1011
\@@glossarysecstar 10, 10, 11, 52, 53, 228, 379, 402, 504, 629, 652, 755, 756, 883, 908, 1011
\@@gls@@default@acr@descplural 261, 776
\@@gls@@default@acr@firstplural 261, 776
\@@gls@@default@acr@plural 261, 776
\@@gls@checkactual 132, 452, 703, 956, 957
\@@gls@checkbar . . 131, 451, 702, 956
\@@gls@checkescactual 129, 130, 449, 450, 700, 701, 954
\@@gls@checkescbar 130, 450, 701, 955
\@@gls@checkesclevel 130, 131, 450, 451, 701, 702, 955
\@@gls@checkescquote 129, 189, 449, 480, 481, 700, 732, 733, 954, 987
\@@gls@checklevel 131, 132, 451, 452, 702, 703, 956
\@@gls@checkquote 128, 129, 187, 448, 449, 479, 699, 700, 730, 731, 953, 985
\@@gls@codepage . . 191, 192, 988–990
\@@gls@default@entryfmt . . 349, 437, 443, 444, 548, 688, 694, 801
\@@gls@expand@field 24, 82, 82, 88, 386, 418, 419, 422, 524–527, 529, 531, 533, 535, 569–573, 636, 669, 672, 673, 776, 778, 779, 781, 783, 784, 786, 788, 823–826, 890, 925, 928, 929
\@@gls@extramakeindexopts 187, 194, 195, 478, 484, 730, 736, 985, 991
\@@gls@fixbraces 211, 211, 493, 745, 1000, 1001
\@@gls@navhypertarget 273, 273, 1033
\@@gls@noexpand@assign@default@plurals 261, 776, 823
\@@gls@noexpand@field 24, 81, 82, 386, 418, 636, 668, 669, 890, 924, 925
\@@gls@noidx@no@sanitizesort 26, 387, 637, 891

\@@gls@noidx@nosanitizesort 26, 199, 387, 487, 637, 739, 891, 994
\@@gls@noidx@sanitizesort 25, 387, 636, 891
\@@gls@nosanitizesort 25, 25, 199, 386, 387, 487, 636, 739, 890, 891, 994
\@@gls@sanitizesort 25, 25, 199, 386, 387, 487, 636, 739, 890, 891, 994
\@@gls@xdycheckbackslash 134, 453, 704, 705, 958
\@@gls@xdycheckquote 133, 452, 704, 957, 958
\@@glslocalreset 106, 107, 109, 433, 435, 684, 685, 941, 942
\@@glslocalunset 106, 106, 109, 433, 434, 684, 685, 941, 942
\@@glsreset 107, 107, 109, 433, 434, 684, 685, 941, 942
\@@glsshowtarget 7, 8, 9, 377, 378, 627, 628, 881, 882
\@@glsunset 106, 106, 109, 433, 434, 684, 685, 941, 942
\@@newglossaryentry@defcounters 108, 108, 434, 685, 942
\@@this@glo@ 64, 409, 659, 915
\@ACRfull 245, 512, 764, 1019
\@ACRfullpl . . 246, 512, 513, 764, 1019
\@ACRlong 163, 244, 357, 466, 512, 556, 717, 764, 809, 971, 1018
\@ACRlongpl 165, 244, 467, 513, 718, 765, 972, 1018
\@ACRshort . . . 159, 244, 356, 463, 512, 555, 715, 764, 809, 969, 1018
\@ACRshortpl 161, 244, 464, 465, 513, 716, 765, 970, 1018
\@Acrfull 244, 512, 764, 1019
\@Acrfullpl 245, 512, 764, 1019
\@Acrlong 162, 243, 356, 465, 512, 556, 717, 764, 809, 971, 1018
\@Acrlongpl 164, 244, 466, 512, 718, 764, 972, 1018
\@Acrshort 158, 159, 355, 463, 555, 715, 808, 969
\@Acrshortpl 160, 464, 716, 970
\@Alph 204, 207, 208, 490, 491, 742, 743, 997–999
\@GLS 138, 139, 455, 706, 707, 960
\@GLS@ 139, 139, 335, 455, 595, 707, 848, 960
\@GLSdesc 148, 149, 459, 711, 964
\@GLSdesc@ 149, 359, 459, 558, 711, 811, 964
\@GLSdescplural 149, 150, 460, 711, 965
\@GLSdescplural@ 150, 359, 460, 558, 711, 811, 965
\@GLSfirst . . . 144, 145, 458, 709, 963
\@GLSfirst@ 145, 358, 458, 557, 709, 810, 963
\@GLSfirstplural 146, 147, 458, 710, 964
\@GLSfirstplural@ 147, 358, 459, 557, 710, 811, 964
\@GLSname 147, 148, 459, 710, 964
\@GLSname@ 148, 359, 459, 558, 711, 811, 964
\@GLSpl 141, 141, 456, 708, 961
\@GLSpl@ 141, 336, 456, 596, 708, 849, 961
\@GLSplural . . . 145, 146, 458, 710, 963
\@GLSplural@ 146, 358, 458, 557, 710, 810, 963
\@GLSsymbol . . . 150, 151, 460, 712, 965
\@GLSsymbol@ 151, 360, 460, 558, 712, 811, 965
\@GLSsymbolplural 151, 152, 460, 712, 966
\@GLSsymbolplural@ 152, 360, 460, 558, 712, 812, 966
\@GLStext 143, 457, 709, 962
\@GLStext@ 143, 357, 457, 557, 709, 810, 962
\@GLSuseri . . . 152, 153, 461, 712, 966
\@GLSuseri@ 153, 360, 461, 559, 712, 812, 966
\@GLSuserii . . . 153, 154, 461, 713, 966
\@GLSuserii@ 154, 361, 461, 559, 713, 812, 966
\@GLSuseriii . . 154, 155, 461, 713, 967
\@GLSuseriii@ 155, 361, 461, 559, 713, 812, 967
\@GLSuseriv . . . 155, 156, 462, 713, 967
\@GLSuseriv@ 156, 361, 462, 559, 713, 812, 967
\@GLSuserv . . . 156, 157, 462, 714, 968
\@GLSuserv@ 157, 362, 462, 560, 714, 813, 968
\@GLSuservi . . . 157, 158, 462, 714, 968

\@GLSuservi@ 158, 362, 462, 560, 714, 813, 968
\@Gls 138, 454, 706, 960
\@Gls@ 110, 112, 138, 138, 334, 435, 436, 454, 594, 686, 687, 706, 847, 943, 944, 960
\@Gls@@link 122, 949
\@Gls@acrentryname 168, 247, 468, 513, 719, 765, 974, 1020
\@Gls@entry@field 166, 167–169, 423, 467–471, 673, 719–722, 973, 974
\@Gls@entryname . . 167, 167, 247, 468, 513, 719, 765, 973, 974, 1020
\@GlsSetXdyFirstLetterAfterDigits 179, 180, 473, 725, 979, 980
\@GlsSetXdyNumberGroupOrder 180, 473, 474, 725, 980
\@Glsdesc 148, 459, 711, 964
\@Glsdesc@ 148, 359, 459, 558, 711, 811, 964
\@Glsdescplural . . 149, 459, 711, 965
\@Glsdescplural@ 149, 359, 459, 558, 711, 811, 965
\@Glsdisp 142, 142, 962
\@Glsentryfield 89, 167, 169–174, 330, 331, 929, 973–977
\@Glsfirst 144, 458, 709, 963
\@Glsfirst@ 144, 358, 458, 557, 709, 810, 963
\@Glsfirstplural . 146, 458, 710, 963
\@Glsfirstplural@ 146, 358, 458, 710, 963
\@Glsname 147, 459, 710, 964
\@Glsname@ 147, 359, 459, 557, 710, 811, 964
\@Glspl 140, 456, 707, 961
\@Glspl@ 110, 113, 140, 140, 335, 435, 437, 456, 594, 595, 686, 687, 707, 848, 943, 945, 961
\@Glsplural 145, 458, 710, 963
\@Glsplural@ 145, 358, 458, 557, 710, 810, 963
\@Glssymbol 150, 460, 711, 965
\@Glssymbol@ 150, 360, 460, 558, 711, 811, 965
\@Glssymbolplural 151, 460, 712, 965
\@Glssymbolplural@ 151, 360, 460, 558, 712, 812, 965
\@Glstext 143, 144, 457, 709, 962
\@Glstext@ 144, 357, 457, 557, 709, 810, 962
\@Glsuser@i 360–362, 559, 560, 812, 813
\@Glsuseri 152, 461, 712, 966
\@Glsuseri@ . . . 152, 360, 461, 712, 966
\@Glsuserii 153, 461, 713, 966
\@Glsuserii@ . . 153, 361, 461, 713, 966
\@Glsuseriii 154, 461, 713, 967
\@Glsuseriii@ . 154, 361, 461, 713, 967
\@Glsuseriv 155, 462, 713, 967
\@Glsuseriv@ . . 155, 361, 462, 713, 967
\@Glsuserv 156, 462, 714, 967
\@Glsuserv@ . . . 156, 362, 462, 714, 967
\@Glsuservi 157, 462, 714, 968
\@Glsuservi@ . . 157, 362, 462, 714, 968
\@Mi 292, 605, 859
\@PGLS 335, 335, 595, 848
\@PGLS@ 335, 335, 595, 848
\@PGLSpl 335, 336, 595, 848
\@PGLSpl@ 336, 336, 595, 848
\@Pgls 333, 333, 594, 847
\@Pgls@ 333, 333, 594, 847
\@Pglspl 334, 334, 594, 847
\@Pglspl@ 334, 334, 594, 847
\@Roman 204, 207, 208, 490, 491, 742, 743, 997, 998
\@acrfull . . . 244, 244, 511, 763, 1018
\@acrfullpl 245, 512, 764, 1019
\@acrlong 162, 243, 356, 465, 511, 556, 716, 717, 763, 809, 970, 971, 1018
\@acrlongpl 164, 244, 466, 512, 718, 764, 972, 1018
\@acrshort 158, 243, 355, 463, 511, 512, 555, 714, 763, 764, 808, 968, 1018
\@acrshortpl 160, 244, 464, 512, 715, 764, 969, 1018
\@addtoacronymlists 21, 384, 634, 888
\@addtoacronynlists 21
\@afterheading 280, 582, 599, 835, 853
\@alph 204, 207, 208, 490, 491, 742, 743, 997–999
\@arabic 204, 207, 208, 490, 491, 742, 743, 997, 998
\@auxout 71, 73, 111, 135, 194, 195, 197, 200, 212, 217, 273, 413, 436, 484–486, 488, 494, 496, 596, 663, 664, 686, 687, 736–738, 740, 746, 748,

849, 919, 920, 943, 944, 991, 993, 995, 1001, 1003, 1004, 1033
\@backslashchar .. 127, 128, 133, 134, 188, 447, 453, 698, 699, 704, 705, 732, 952, 953, 958, 986
\@bsphack 202, 489, 741, 996
\@cGls 111, 112, 436, 687, 944
\@cGls@ 110, 112, 112, 435, 436, 686, 687, 943, 944
\@cGlspl 112, 113, 436, 687, 944
\@cGlspl@ 110, 113, 113, 435–437, 686, 687, 943, 944
\@cclv 292, 605, 606, 859
\@cgls 111, 111, 436, 687, 944
\@cgls@ 109, 111, 111, 435, 436, 685, 687, 942, 944
\@cglspl . 112, 112, 113, 436, 687, 944
\@cglspl@ 109, 112, 112, 435, 436, 686, 687, 943, 944
\@chapter 43, 397, 647, 903
\@classoptionslist . 41, 396, 646, 902
\@closegls 191, 191, 192, 481, 482, 733, 734, 988, 989
\@colht 292, 606, 859
\@colroom 292, 606, 859
\@currentlabelname 11, 228, 379, 504, 629, 756, 883, 1011
\@curroptions 41, 396, 646, 902
\@declaredoptions . 41, 396, 646, 902
\@delimN 510, 762
\@delimR 510, 762
\@disable@onlypremakeg 43, 195, 397, 484, 648, 736, 903, 991
\@disable@premakecs 43, 43, 398, 648, 903
\@disabled@glsaddxdycounters 55, 57, 404, 654, 655, 910
\@do@addcounter 55, 403, 404, 654, 909
\@do@auxoutstuff 217, 496, 748, 1003, 1004
\@do@glossentry .. 506, 540, 758, 793
\@do@gls@checkesc 128, 699, 953
\@do@gls@getcounterprefix 209, 490, 491, 742, 743, 999, 1000
\@do@gls@islistofacronyms 21, 384, 634, 888
\@do@glssee 100, 430, 681, 937
\@do@ifinlist 55, 403, 653, 909
\@do@newglossaryentry 247, 513, 523–536, 569–573, 765, 775–790, 822–826, 1020
\@do@seeglossary 197, 210, 211, 485, 493, 737, 744, 745, 993, 1000, 1001
\@do@subglossentry 507, 540, 759, 793
\@do@wrglossary .. 124, 203, 446, 489, 576, 696, 741, 829, 950, 996
\@do@write@glslabels 35, 36, 393, 643, 897, 898
\@do@writeaux@info 212, 494, 746, 1001
\@doendpe 216
\@domakeglossaries ... 34, 34, 35, 194, 196, 392, 393, 484, 485, 642, 643, 736, 737, 897, 991, 992
\@ehc 292, 605, 859
\@empty 14, 18, 21, 38, 40, 41, 43, 55, 56, 60, 62, 63, 97, 102, 125, 137–141, 158–165, 181, 184, 186, 191–193, 201, 202, 210, 236, 355–357, 381, 383, 384, 394, 396, 397, 403, 404, 406, 408, 409, 428, 431, 446, 454–456, 463–467, 474, 476, 478, 482, 483, 488, 489, 492, 508, 524, 526–533, 535, 537, 555, 556, 576, 578–580, 630, 632, 634, 644, 646, 647, 653, 654, 657, 659, 678, 679, 682, 697, 706–708, 714–719, 726, 728, 730, 733–735, 740, 741, 744, 760, 776, 778, 780, 782, 784–786, 788, 790, 808–810, 830, 832, 833, 884, 886–888, 899–903, 909, 910, 912, 914, 915, 934, 935, 938, 950, 951, 959–961, 968–973, 980, 983, 984, 988–990, 995, 996, 1000, 1015
\@end@fixbraces 211, 493, 745, 1000, 1001
\@endfortrue 31, 67, 87, 391, 411, 421, 596, 641, 661, 672, 849, 895, 916, 928, 1033
\@esphack 203, 489, 741, 996
\@expandtwoargs ... 41, 396, 646, 902
\@firstofone 26, 35, 278, 387, 388, 393, 637, 643, 852, 891, 892, 897
\@firstofthree 121, 136, 137, 140, 142, 158, 160, 162, 164, 355–357, 444, 454, 455, 457, 463–

466, 555, 556, 695, 706–708, 714, 716–718, 808–810, 949, 959, 961, 962, 968, 969, 971, 972
\@firstoftwo 29–31, 86, 87, 104, 121, 140, 141, 160, 161, 164, 165, 390, 391, 420, 421, 444, 455, 456, 464–467, 639–641, 671, 672, 695, 707, 708, 715, 716, 718, 719, 894, 895, 927, 928, 939, 949, 961, 969, 970, 972, 973
\@for . 31, 41, 43, 45, 55, 56, 63, 64, 86, 87, 127, 181–183, 194–196, 203, 211, 218, 248, 391, 396, 397, 403, 404, 409, 420, 421, 447, 474, 475, 484, 485, 490, 493, 497, 514, 524, 526, 528, 529, 531, 534, 535, 537, 576, 596, 597, 641, 646, 648, 654, 659, 671, 672, 698, 726, 727, 736, 737, 742, 745, 748, 749, 766, 776, 779, 780, 782, 784, 787, 788, 790, 830, 849, 850, 895, 902–904, 909, 910, 915, 927, 928, 952, 980–982, 990–992, 997, 1001, 1004, 1021, 1033
\@glo@@desc 99, 429, 680, 936
\@glo@@symbol 99, 430, 681, 937
\@glo@access . . 338, 341, 346, 540, 542, 543, 545, 546, 794–796, 799
\@glo@addchildren 218, 218, 223, 497, 500, 749, 752, 1004, 1007
\@glo@assign@sortkey 195, 198, 229, 485, 486, 504, 736, 738, 756, 992, 993, 1012
\@glo@autosee . 100, 100, 430, 681, 937
\@glo@autoseehook 100, 101, 430, 681, 937
\@glo@check@mkidxrangechar 126, 209, 447, 492, 576, 698, 743, 829, 951, 999
\@glo@check@sortallowed 16, 16, 17–19, 196, 199, 381–383, 485, 487, 631–633, 737, 739, 885–887, 992, 994
\@glo@childlist . 218, 497, 749, 1004
\@glo@counter 77, 95, 99, 416, 427, 429, 666, 678, 680, 922, 934, 936
\@glo@counterprefix 200, 205, 208–210, 236, 239, 488, 490–492, 508, 510, 740, 742–744, 760, 762, 995, 998–1000, 1015, 1017
\@glo@default@sorttype 15, 15, 198, 221, 222, 381, 486, 499, 500, 631, 738, 750–752, 885, 993, 1006, 1007
\@glo@defaultcounter 98, 99, 429, 680, 936
\@glo@desc 75, 94, 95, 97, 99, 415, 426–430, 665, 677, 678, 680, 921, 933, 934, 936
\@glo@descaccess 339, 341, 342, 541–543, 794–796
\@glo@descplural 75, 94, 95, 415, 426, 427, 665, 677, 921, 933, 934
\@glo@descpluralaccess 339, 341, 342, 541–543, 794–796
\@glo@do@sortentries 218, 219, 497, 748, 749, 1004, 1005
\@glo@entry 178, 473, 724, 979
\@glo@entryprefix 329, 330, 591, 592, 844, 845
\@glo@entryprefixfirst 329, 330, 591, 592, 844, 845
\@glo@entryprefixfirstplural . . 329, 330, 591, 592, 844, 845
\@glo@entryprefixplural 329, 330, 591, 592, 844, 845
\@glo@esclabel 102, 103, 431, 432, 682, 683, 938, 939
\@glo@etext 437–439, 688–690
\@glo@first . . . 76, 95, 98, 99, 415, 427, 429, 430, 533, 572, 665, 678, 680, 786, 826, 921, 934, 936
\@glo@firstaccess 338, 341, 541–543, 794–796
\@glo@firstplural 76, 95, 98, 415, 427, 429, 572, 665, 678, 680, 826, 921, 934, 936
\@glo@firstpluralaccess 338, 341, 541–543, 794–796
\@glo@grabfirst 224, 224, 501, 753, 1008
\@glo@label 78, 80, 88, 89, 91–100, 108, 176, 177, 324, 330, 341, 342, 417, 422, 424–430, 434, 471, 472, 542–544, 592, 623, 624, 667, 668, 673, 675–681, 685, 723,

724, 796, 797, 845, 877, 922–924, 929, 931–937, 942, 978, 979
\@glo@list 100, 430, 681, 937
\@glo@long 68, 80, 96, 99, 411, 417, 427, 429, 661, 667, 678, 680, 917, 923, 934, 936
\@glo@longaccess 339, 341, 342, 541–543, 794–796
\@glo@longpl 80, 96, 99, 417, 427, 429, 523, 525, 527, 529, 530, 533, 534, 569, 667, 678, 680, 775, 776, 778, 779, 781, 783, 786, 787, 822, 923, 934, 936
\@glo@longpluralaccess 339, 341, 342, 541–543, 794–796
\@glo@name 16, 75, 81, 95, 97–99, 382, 415, 418, 427–429, 631, 665, 668, 677, 679, 680, 885, 921, 924, 933, 935, 936
\@glo@no@assign@sortkey . . 195, 229, 485, 504, 736, 756, 992, 1012
\@glo@nonumberlist 79, 416, 417, 667, 923
\@glo@numfmt 209, 492, 576, 744, 829, 999
\@glo@parent 18, 78, 95, 97, 98, 102, 103, 219, 383, 416, 427, 428, 431, 432, 497, 632, 666, 678, 679, 682, 683, 749, 886, 922, 934, 935, 938, 939, 1005
\@glo@plural 76, 95, 97, 98, 415, 427–429, 571, 665, 678–680, 824, 921, 934–936
\@glo@pluralaccess 338, 341, 541–543, 794–796
\@glo@prefix 12, 78, 96, 102, 103, 126, 209, 380, 416, 427, 431, 432, 447, 492, 576, 629, 666, 678, 682, 683, 698, 744, 829, 884, 922, 934, 938, 939, 951, 952, 999
\@glo@range 209, 492, 576, 744, 829, 999
\@glo@see . . 77, 95, 100, 416, 427, 430, 666, 678, 681, 922, 934, 937
\@glo@seeautonumberlist . . . 12, 12, 77, 380, 416, 629, 666, 883, 922
\@glo@short . . . 68, 80, 96, 99, 411, 417, 427, 429, 572, 662, 667, 678, 680, 826, 917, 923, 934, 936
\@glo@shortaccess 339, 341, 342, 541–543, 569–572, 794–796, 822–824, 826
\@glo@shortpl 80, 96, 99, 417, 427, 429, 523, 525, 527, 529, 530, 533, 534, 569, 572, 667, 678, 680, 775–777, 779, 781, 783, 786, 787, 822, 826, 923, 934, 936
\@glo@shortpluralaccess 339, 341, 342, 541–543, 794–796
\@glo@sort 16, 18, 25, 26, 76, 95, 98, 102, 103, 382, 383, 387, 415, 427, 429, 431, 432, 631, 633, 636, 637, 665, 678, 679, 682, 683, 885–887, 891, 921, 934, 935, 938, 939
\@glo@sortedinsert 219, 219, 497, 498, 749, 750, 1005
\@glo@sortentries 218, 221, 222, 496, 499, 500, 748, 750, 751, 1004, 1006, 1007
\@glo@sorthandler@case . . . 220, 222, 498, 499, 750, 751, 1005, 1007
\@glo@sorthandler@letter . . 220, 221, 498, 499, 750, 751, 1005, 1006
\@glo@sorthandler@nocase . . 220, 222, 498, 500, 750, 751, 1006, 1007
\@glo@sorthandler@word 220, 221, 498, 499, 750, 1005, 1006
\@glo@sortinghandler 218, 219, 496, 498, 748, 750, 1004, 1005
\@glo@sortinglist 218, 219, 222, 496–498, 500, 748, 749, 752, 1004, 1005, 1007
\@glo@sortmacro@case 221, 499, 751, 1007
\@glo@sortmacro@def 222, 500, 752, 1007
\@glo@sortmacro@def@do 222, 500, 752, 1007
\@glo@sortmacro@letter 221, 499, 751, 1006
\@glo@sortmacro@nocase 222, 499, 751, 1007
\@glo@sortmacro@standard 221, 499, 751, 1006
\@glo@sortmacro@use 223, 500, 752, 1008
\@glo@sortmacro@word 221, 498, 750, 1006

\@glo@sorttype 198, 223, 224, 229, 486, 500, 501, 504, 738, 752, 753, 756, 993, 1008, 1012
\@glo@storeentry 16–18, 102, 381–383, 431, 631, 632, 682, 885–887, 938
\@glo@suffix . . 126, 209, 447, 492, 576, 698, 744, 829, 951, 952, 999
\@glo@symbol 68, 76, 95, 99, 100, 411, 415, 427, 430, 528, 532, 661, 665, 678, 681, 781, 785, 917, 921, 934, 937
\@glo@symbolaccess 338, 341, 541–543, 572, 794–796, 825
\@glo@symbolplural 77, 95, 100, 415, 427, 430, 665, 678, 681, 921, 934, 937
\@glo@symbolpluralaccess 338, 341, 342, 541–543, 794–796
\@glo@text 76, 95, 98, 99, 137–142, 166, 168, 169, 415, 427, 429, 430, 454–457, 467, 468, 529, 571, 592, 665, 678, 679, 681, 706–708, 719, 720, 781, 824, 845, 921, 934, 935, 937, 960–962, 973, 974
\@glo@textaccess 338, 341, 540, 542, 543, 569–571, 794–796, 822–825
\@glo@thislabel 101, 431, 681, 682, 938
\@glo@thislettergrp 224, 225, 501, 502, 753, 754, 1008, 1009
\@glo@thisvalue 69, 70, 412, 662, 918
\@glo@tmp 71, 88, 89, 210, 422, 492, 673, 744, 919, 929, 1000
\@glo@type 11, 18, 77, 95–100, 177, 178, 195, 201, 214–219, 222–224, 227, 228, 247, 379, 383, 415, 427–430, 472, 473, 484, 488, 489, 495–497, 500, 501, 503, 504, 513, 524, 526, 527, 529, 531, 533, 535, 537, 596, 597, 629, 632, 665, 677–681, 724, 736, 740, 741, 746–749, 752, 753, 755, 756, 765, 776, 778, 780, 782, 784, 786, 788, 790, 849, 850, 883, 886, 887, 921, 934–937, 979, 991, 995, 996, 1002–1005, 1007, 1008, 1010, 1011, 1020
\@glo@types 35, 63, 64, 70, 72, 107, 108, 178, 193, 195, 196, 269, 324, 393, 409, 413, 433, 434, 473, 484, 485, 539, 623, 643, 659, 663, 684, 685, 724, 736, 737, 792, 877, 898, 915, 918, 919, 941, 942, 979, 990–992, 1032
\@glo@useri . . 79, 96, 99, 417, 427, 429, 667, 678, 680, 923, 934, 936
\@glo@useriaccess 339, 341, 342, 541–543, 794–796
\@glo@userii . 79, 96, 99, 417, 427, 429, 667, 678, 680, 923, 934, 936
\@glo@useriiaccess 339, 341, 342, 541–543, 794, 795, 797
\@glo@useriii 79, 96, 99, 417, 427, 429, 667, 678, 680, 923, 934, 936
\@glo@useriiiaccess 339, 341, 342, 541–543, 794, 795, 797
\@glo@useriv . 79, 96, 99, 417, 427, 429, 667, 678, 680, 923, 934, 936
\@glo@userivaccess 340–342, 541–543, 795–797
\@glo@userv . . 79, 96, 99, 417, 427, 429, 667, 678, 680, 923, 934, 936
\@glo@uservaccess 340–342, 541, 542, 544, 795–797
\@glo@uservi . 80, 96, 99, 417, 427, 429, 667, 678, 680, 923, 934, 936
\@glo@uservivaccess 340–342, 541, 542, 544, 795–797
\@glodesc 99, 430, 680, 937
\@glolist@ 97, 428, 679, 935
\@gloname 99, 429, 680, 936
\@glossary@default@style 11, 11, 13, 214, 237, 265, 379, 380, 495, 509, 537, 629, 630, 747, 761, 790, 883, 884, 1002, 1015, 1030, 1031
\@glossaryentryfield 101, 102, 103, 431, 432, 682, 938, 939
\@glossarysection 51, 52, 401, 402, 651, 652, 907
\@glossarystyle 214, 215, 228, 495, 503, 747, 755, 1002, 1003, 1011
\@glossarysubentryfield 101, 102, 103, 431, 432, 682, 683, 938, 939
\@gls 137, 137, 454, 706, 959

\@gls@ 109, 111, 137, 137, 332, 334, 435, 436, 454, 593, 594, 686, 687, 706, 846, 847, 943, 944, 959
\@gls@@automake@immediate 193, 194, 484, 735, 736, 990, 991
\@gls@@link 122, 122, 444, 445, 695, 949
\@gls@Hcounter 125, 446, 697, 950, 951
\@gls@ReturnAfterFi 240, 510, 511, 762, 1017
\@gls@access@display . . 346, 545, 799
\@gls@actualchar 103, 129, 130, 132, 185, 432, 449, 450, 452, 477, 579, 682, 683, 700, 701, 703, 728, 729, 832, 938, 939, 954, 956, 957, 983
\@gls@addpredefinedattributes . . 57, 181, 190, 405, 474, 481, 655, 726, 733, 910, 911, 980, 988
\@gls@adjustmode 177, 178, 472, 724, 979
\@gls@after . . . 21, 384, 385, 634, 889
\@gls@assign@symbol@field 25
\@gls@assign@symbolplural@field 25
\@gls@automake 191, 196, 481, 482, 485, 733, 734, 737, 988, 989, 992
\@gls@automake@immediate . . 192, 194, 482–484, 734–736, 989, 990
\@gls@automake@types 193, 194, 196, 990, 992
\@gls@before 21, 384, 634, 889
\@gls@between 274, 596, 597, 850, 1033, 1034
\@gls@body . . . 168, 468, 719, 720, 974
\@gls@checkactual 128, 132, 188, 448, 452, 479, 699, 703, 731, 953, 956, 957, 986
\@gls@checkbar 128, 131, 188, 448, 451, 480, 699, 702, 731, 953, 955, 956, 986
\@gls@checkedmkidx 127–134, 187–189, 447–453, 478–480, 698–705, 730–732, 952–958, 985–987
\@gls@checkesc 128, 132, 188, 699, 732, 953, 986
\@gls@checkescactual 127, 129, 188, 448–450, 479, 699–701, 731, 952, 954, 986
\@gls@checkescbar 128, 130, 188, 448, 450, 480, 699, 701, 731, 953–955, 986
\@gls@checkesclevel 128, 130, 188, 450, 451, 699, 701, 702, 732, 953, 955, 986
\@gls@checkescquote 127, 129, 188, 189, 448, 449, 479–481, 699, 700, 731, 732, 952–954, 986, 987
\@gls@checklevel 128, 131, 188, 448, 451, 480, 699, 702, 703, 731, 953, 956, 986
\@gls@checkmkidxchars 102, 126, 127, 188, 197, 208, 210, 431, 447, 448, 479, 485, 491, 493, 576, 682, 698, 699, 731, 737, 743, 744, 829, 938, 951, 952, 986, 992, 999, 1000
\@gls@checkquote . 127, 128, 187, 188, 448, 449, 478, 479, 699, 700, 730, 731, 952, 953, 985, 986
\@gls@classI 181, 182, 474, 475, 726, 981
\@gls@classII 182, 475, 726, 981
\@gls@codepage 62, 217, 408, 496, 658, 748, 914, 1004
\@gls@counter 120, 123, 125, 126, 177, 200, 209, 210, 444–447, 472, 488, 492, 576, 695–697, 724, 740, 744, 829, 948, 950, 951, 979, 995, 999, 1000
\@gls@counterwithin 14, 15, 381, 630, 631, 884, 885
\@gls@ctr 55, 403, 654, 909
\@gls@currentlettergroup . . 223, 225, 500, 502, 752, 754, 1008, 1009
\@gls@debugfalse . . . 7, 377, 627, 881
\@gls@debugtrue 7, 377, 626, 627, 880, 881
\@gls@declareoption 11, 12, 13, 20, 23, 24, 30, 33, 34, 36, 37, 40, 41, 379, 380, 383–386, 390, 392, 393, 395, 396, 629, 630, 633–636, 640, 642, 643, 645, 646, 883, 884, 887–890, 894, 896–899, 901
\@gls@default 113, 437, 688, 945
\@gls@default@value 68–70, 76, 81, 82, 95, 97–99, 263, 330, 411, 412, 415, 418, 419, 427–430, 532, 591, 592, 661, 662, 665, 668,

669, 677–680, 785, 845, 917, 918, 921, 924, 925, 934–937, 1030
\@gls@defaulttype 71, 71, 919
\@gls@deffile 84–86, 420, 421, 670, 671, 926, 927
\@gls@define@glossaryentrycounter . . . 14, 14, 42, 229, 230, 381, 397, 504, 505, 630, 631, 647, 756, 757, 884, 885, 902, 1011, 1012
\@gls@define@glossarysubentrycounter 15, 42, 229, 230, 381, 397, 504, 505, 631, 647, 756, 757, 885, 903, 1011, 1012
\@gls@defsort 16–19, 99, 382, 383, 430, 631–633, 681, 885–887, 937
\@gls@defsortcount 16–19, 73, 382, 383, 414, 631–633, 664, 885–887, 920
\@gls@deprecated@acr@warning 263, 778, 780, 782, 784, 786, 788, 1030
\@gls@do@acronymsdef 20, 20, 42, 74, 383, 384, 397, 414, 633, 634, 647, 664, 888, 902, 903, 920
\@gls@do@automake@err 36, 37, 38, 898, 899
\@gls@do@glsunset 120, 120, 137–142, 948, 960–962
\@gls@do@indexdef 40–43, 74, 396, 397, 414, 646, 647, 664, 901–903, 920
\@gls@do@numbersdef 40, 42, 74, 395–397, 414, 646, 647, 664, 901–903, 920
\@gls@do@symbolsdef 40, 74, 395, 414, 645, 664, 901, 920
\@gls@do@symbolssdef 42, 397, 647, 902, 903
\@gls@doautomake 37, 37, 38, 194, 196, 394, 484, 485, 644, 736, 737, 899, 991, 992
\@gls@docheckquotedef 187, 189, 478–481, 730–733, 985–987
\@gls@docloadedfalse 6, 376, 626, 880
\@gls@docloadedtrue . 6, 376, 626, 880
\@gls@dodeflistparser 195, 485, 736, 991, 992
\@gls@doentrycounterdef 42, 43, 397, 647, 902, 903
\@gls@doentrydef . 443, 444, 694, 695
\@gls@dolast . 211, 212, 493, 745, 1001
\@gls@donext . 211, 212, 493, 745, 1001
\@gls@donext@def 176, 472, 723, 724, 978
\@gls@dosubentrycounterdef 42, 43, 397, 647, 903
\@gls@dothiswrite 191–193, 482, 483, 734, 735, 988–990
\@gls@elem 596, 849, 1033
\@gls@enablesavenonumberlist 79, 85, 416, 420, 667, 671, 923, 927
\@gls@encapchar 130, 131, 185, 209, 211, 450, 451, 477, 492, 493, 576, 579, 701, 702, 728, 729, 744, 745, 829, 832, 955, 956, 983, 999, 1000
\@gls@entry@count 110, 111, 111, 435, 436, 686, 687, 943, 944
\@gls@entry@field 88, 89, 109, 135, 166, 167–175, 342–344, 422, 423, 434, 467–471, 544, 545, 673, 685, 719–723, 797, 798, 929, 942, 973–977
\@gls@escbsdq 127, 127, 186, 447, 448, 478, 580, 698, 699, 729, 730, 833, 952, 984
\@gls@expand@fields 82, 82, 83, 419, 669, 925
\@gls@expandonce . . 82, 419, 669, 925
\@gls@extramakeindexopts . . 189, 195, 481, 484, 733, 736, 987, 991
\@gls@fetchfield 69, 87, 412, 421, 662, 672, 918, 928
\@gls@field@link . . . 89, 90, 143, 143, 144–158, 357–362, 423, 424, 457–463, 557–560, 674, 709–714, 810–813, 930, 962–968
\@gls@fieldaccess@display 346, 347–349, 546–548, 799–801
\@gls@firsttok . . 224, 501, 753, 1008
\@gls@fixbraces 100, 211, 430, 493, 681, 745, 937, 1000
\@gls@forbidtexext 71, 73, 413, 663, 664, 919
\@gls@fulllist 35, 643, 898
\@gls@get@counterprefix 210, 492, 744, 1000
\@gls@getbody 168, 468, 719, 974
\@gls@getcounter . . 74, 414, 664, 920

\@gls@getcounterprefix 209, 210, 490–492, 742–744, 1000
\@gls@getgrouptitle 197, 235, 235, 274, 485, 508, 597, 737, 760, 850, 993, 1014, 1034
\@gls@glossary 202, 202, 489, 741, 996
\@gls@glslink@hyper@update@hook 120, 120, 948
\@gls@gobbleopt . . . 72, 413, 663, 919
\@gls@grptitle 235, 271, 274, 508, 596, 597, 760, 849, 850, 1014, 1033, 1034
\@gls@hyp@opt 89, 90, 111, 112, 121, 122, 137–165, 244–246, 332–335, 423, 424, 436, 444, 454–467, 511, 512, 593–595, 674, 687, 695, 706–718, 763, 764, 846–848, 930, 944, 949, 959–972, 1018, 1019
\@gls@hyp@opt@cs . 121, 444, 695, 949
\@gls@hypergroup 273, 274, 596, 849, 1033
\@gls@if@measuring 103, 103, 939
\@gls@ifinlist 55, 55, 403, 653, 654, 909
\@gls@ifnotmeasuring 432, 683
\@gls@igtype 75, 415, 665, 921
\@gls@increment@currcount . 109, 110, 434, 435, 685, 686, 942, 943
\@gls@indexdef . 40, 41, 396, 646, 901
\@gls@initnonumberlist 78, 79, 79, 96, 416, 427, 666, 667, 678, 922, 923, 934
\@gls@islistofacronyms 21, 384, 634, 888
\@gls@keylist 373, 569, 822
\@gls@keymap 79, 86, 86, 87–89, 330, 340, 417, 420–422, 542, 591, 667, 671–673, 795, 844, 923, 927–929
\@gls@label 197, 200, 206, 208, 209, 485, 486, 488, 491, 492, 737, 740, 742–744, 993, 995, 998, 999
\@gls@labelsfile . . 35, 393, 643, 898
\@gls@langmod 33, 191, 192, 217, 482, 483, 734, 735, 896, 988, 989, 1003
\@gls@levelchar 103, 130–132, 185, 432, 450, 451, 477, 579, 683, 701, 702, 728, 729, 832, 939, 955, 956, 983
\@gls@link . . . 122, 123, 137–143, 158–165, 355–357, 445, 454–457, 463–467, 555, 556, 695, 696, 706–709, 714–719, 808–810, 949, 950, 960–962, 968–973
\@gls@link@checkfirsthyper 122, 137–142, 445, 454–457, 696, 706–708, 949, 959–962
\@gls@link@label 123, 445, 525, 530, 696, 777, 783, 950
\@gls@link@nocheckfirsthyper . . 142, 143, 158–165, 457, 463–467, 709, 714–718, 962, 968–973
\@gls@link@opts 123, 445, 525, 530, 696, 777, 783, 950
\@gls@list . . 596, 597, 849, 850, 1033
\@gls@listsuffix . . 55, 403, 653, 909
\@gls@loadlist 13, 13, 264, 380, 537, 630, 790, 884, 1030
\@gls@loadlong . . 12, 12, 13, 264, 380, 537, 629, 630, 790, 884, 1030
\@gls@loadsuper 12, 12, 13, 264, 380, 537, 630, 790, 884, 1030
\@gls@loadtree 13, 13, 264, 380, 537, 630, 790, 884, 1030
\@gls@local@increment@currcount 109, 110, 434, 435, 685, 686, 942, 943
\@gls@loclist 198, 199, 224, 225, 486, 487, 501, 502, 738, 739, 753, 754, 993, 994, 1009
\@gls@map 86, 87, 420, 421, 671, 672, 927, 928
\@gls@missinglang@warn 23, 23, 46, 386, 399, 635, 649, 890, 905
\@gls@missingnumberlist . . . 81, 99, 418, 429, 668, 680, 924, 936
\@gls@noaccess 340, 346, 542, 545, 546, 795, 799
\@gls@noexpand@field 81
\@gls@noexpand@fields 81, 83, 418, 419, 668, 669, 924, 925
\@gls@nohyperlist 22, 22, 74, 75, 123, 385, 414, 415, 445, 635, 665, 696, 889, 921, 950
\@gls@noidx@do 223, 224, 501, 752, 753, 1008, 1009
\@gls@noidx@getgrouptitle . 197, 235, 485, 508, 737, 760, 993, 1014

\@gls@noidx@sanitizesort . . . 25, 199, 387, 487, 636, 739, 891, 994
\@gls@noidx@setsanitizesort 28, 28, 199, 389, 487, 638, 639, 739, 893, 994
\@gls@noidxloclist@finalsep 198, 199, 487, 738, 739, 994
\@gls@noidxloclist@prev 199, 226, 487, 502, 503, 739, 754, 755, 994, 1010
\@gls@noidxloclist@sep 198, 226, 486, 487, 502, 503, 738, 754, 755, 994, 1010
\@gls@noist@toolate 185, 186, 190, 983, 984, 988
\@gls@noref@warn 197, 200, 224, 485, 488, 501, 737, 740, 753, 992, 995, 1008
\@gls@notranslatorhook 29
\@gls@numberlink 239, 510, 762, 1016
\@gls@numbersdef 40, 40, 395, 396, 646, 901
\@gls@numlist@lastsep 176, 472, 723, 724, 978
\@gls@numlist@nextsep 176, 472, 723, 724, 978
\@gls@numlist@sep 176, 472, 723, 724, 978
\@gls@old@chapter . 43, 397, 647, 903
\@gls@oldnewglossaryentryposthook 341, 542, 796
\@gls@oldnewglossaryentryprehook 340, 341, 542, 795
\@gls@onlypremakeg 43, 43, 397, 647, 648, 903
\@gls@order 191–193, 482, 483, 733–735, 988–990
\@gls@org@LT@output . . . 292, 605, 859
\@gls@org@glsnoidxdisplayloc 199, 487, 739, 994
\@gls@org@glsseeformat 199, 487, 739, 994
\@gls@override@glossary . . . 39, 39, 40, 394, 395, 645, 900, 901
\@gls@patchtabularx 105, 432, 683, 940
\@gls@preglossaryhook 214, 216, 494, 495, 746, 747, 1002, 1003
\@gls@prevlevel . . 303, 304, 325–327, 586, 587, 612, 613, 624, 625, 839, 840, 865, 866, 878, 879
\@gls@provide@newglossary 71, 73, 413, 663, 664, 918–920
\@gls@quotechar 128–132, 185, 187, 189, 448–452, 477–480, 579, 700–703, 728–732, 832, 953–957, 983, 985, 987
\@gls@reference 197, 200, 227, 485, 486, 488, 503, 737, 738, 740, 755, 993, 995, 1010
\@gls@removedoption 27, 31, 32, 38, 892, 896, 900
\@gls@removespaces 239, 240, 510, 762, 1016, 1017
\@gls@renewglossary 190, 202, 481, 489, 733, 741, 988, 996
\@gls@replacementtext 345, 346, 545, 798, 799
\@gls@requested@languages 31, 31, 45, 895, 904
\@gls@rest . . . 168, 468, 719, 720, 974
\@gls@restore@glslocal 120, 124, 948, 950
\@gls@restoreat . . . 85, 420, 671, 927
\@gls@roman 58, 405, 406, 577, 656, 830, 911, 912
\@gls@sanitized@tmp 127, 447, 448, 698, 952
\@gls@sanitizedesc 24, 386, 391, 636, 641
\@gls@sanitizename . 25, 386, 636, 890
\@gls@sanitizesort 16, 25, 382, 386, 631, 636, 885, 890
\@gls@sanitizesymbol 25, 386, 391, 392, 636, 641, 890
\@gls@save@glslocal 120, 121, 124, 948, 950
\@gls@saveentrycounter 124, 125, 177, 446, 472, 696, 697, 724, 950, 979
\@gls@savenonumberlist 78, 78, 79, 199, 416, 666, 667, 922, 923, 995
\@gls@see@noindex 9, 9, 78, 378, 379, 416, 628, 666, 882, 922
\@gls@setacrstyle 31, 42, 391, 392, 396, 397, 641, 647, 895, 902, 903
\@gls@setcounter 73, 74, 414, 664, 920
\@gls@setdefault@glslink@opts 123, 124, 445, 696, 950

\@gls@setsort 16–19, 124, 177, 382, 383, 446, 472, 631–633, 696, 724, 886, 887, 950, 979
\@gls@setupshortcuts 42, 396, 397, 647, 902, 903
\@gls@setupsort@clear 18, 887
\@gls@setupsort@def 17, 382, 632, 886
\@gls@setupsort@none 18, 383, 633, 887
\@gls@setupsort@standard 16, 381, 382, 631, 885, 886
\@gls@setupsort@use 17, 382, 632, 886
\@gls@sort 225, 502, 754, 1009
\@gls@sort@A 220, 498, 750, 1005, 1006
\@gls@sort@B 220, 498, 750, 1005, 1006
\@gls@startswithexpandonce 82, 82, 419, 669, 925
\@gls@storenonumberlist . 79, 99, 416, 429, 666, 667, 680, 923, 936
\@gls@symbolsdef 40, 40, 395, 645, 901
\@gls@this . . . 203, 204, 490, 742, 997
\@gls@thisHloc . . 210, 492, 744, 1000
\@gls@thisentryname . . . 35, 643, 898
\@gls@thisfield 69, 70, 412, 662, 918
\@gls@thislabel 67, 211, 212, 222, 410, 493, 500, 661, 745, 752, 916, 1001, 1007
\@gls@thislang 45, 904
\@gls@thislist 176, 472, 723, 724, 978
\@gls@thisloc . . . 210, 492, 744, 1000
\@gls@thisval 87, 421, 672, 928
\@gls@title 51, 401, 651, 907
\@gls@tmp 8, 18, 46, 58, 60, 82, 127, 203, 239, 240, 378, 383, 398, 406, 419, 447, 448, 489, 596, 597, 627, 632, 649, 656, 657, 669, 698, 741, 850, 882, 887, 904, 911, 912, 925, 952, 996, 1017, 1033, 1034
\@gls@tmpb 128, 128, 129–134, 187, 189, 448–453, 478–480, 699–705, 730–732, 953–958, 985, 987
\@gls@toc 52, 53, 54, 402, 403, 652, 653, 908
\@gls@type 194, 196, 248, 324, 484, 485, 514, 524, 526, 528, 529, 531, 534, 535, 537, 623, 736, 737, 766, 776, 779, 780, 782, 784, 787, 788, 790, 877, 990, 992, 1021
\@gls@updatechecked . . . 127, 128, 128, 188, 447, 448, 479, 480, 698, 699, 731, 732, 952, 953, 986
\@gls@usetranslator 29, 30, 31, 45, 390, 391, 398, 639, 640, 648, 894, 895, 904
\@gls@value . 81, 82, 171, 418, 419, 469, 668, 669, 721, 924, 925, 976
\@gls@warnonglossdefined . 24, 24, 213, 386, 494, 636, 746, 890, 1001
\@gls@warnontheglossdefined 24, 24, 232, 386, 506, 636, 758, 890, 1013
\@gls@write@entrycounts 110, 110, 435, 436, 686, 943
\@gls@writedef 84, 85, 86, 420, 670, 671, 926, 927
\@gls@writeisthook 185, 186, 186, 477, 478, 728, 730, 983–985
\@gls@x 239, 240, 762, 1017
\@gls@xdy@Lclass@Alpha-page-numbers 59
\@gls@xdy@Lclass@Appendix-page-numbers 59
\@gls@xdy@Lclass@Roman-page-numbers 58
\@gls@xdy@Lclass@alpha-page-numbers 59
\@gls@xdy@Lclass@arabic-page-numbers 59
\@gls@xdy@Lclass@arabic-section-numbers 59
\@gls@xdy@Lclass@roman-page-numbers 58
\@gls@xdy@locationlist 58, 181, 182, 405, 474, 475, 655, 726, 911, 981
\@gls@xdycheckbackslash 127, 133, 447, 453, 698, 704, 705, 952, 958
\@gls@xdycheckquote . . . 127, 133, 447, 452, 698, 704, 952, 957, 958
\@gls@xref 210, 211, 493, 744, 745, 1000
\@glsAlphacompositor 49, 49, 59, 400, 406, 577, 650, 656, 831, 906, 912
\@glsHlocref 205, 206, 208, 490, 491, 742, 743, 998, 999
\@glsacronymlists 21, 21, 22, 63, 247, 248, 269, 384, 385, 409, 513, 514, 524, 526–529,

531, 533–535, 537, 539, 634, 659, 765, 766, 776, 778–780, 782, 784, 786–788, 790, 792, 888, 889, 915, 1020, 1021, 1032
\@glsaddkey 88, 89, 422, 673, 929
\@glsaddstoragekey 87, 88, 422, 672, 673, 928, 929
\@glsaddxdyattribute 56, 404, 654, 910
\@glsdefaultplural 80, 418, 668
\@glsdefaultsort 16, 81, 382, 418, 631, 668, 885, 924
\@glsdesc 148, 459, 711, 964
\@glsdesc@ 148, 359, 459, 558, 711, 811, 964
\@glsdescplural 149, 459, 711, 964, 965
\@glsdescplural@ 149, 359, 459, 558, 711, 811, 965
\@glsdisp 142, 142, 457, 708, 962
\@glsentry 35, 107, 108, 111, 393, 433, 434, 436, 643, 684–687, 898, 941–944
\@glsentrytitlecase 171, 171, 469, 470, 721, 976
\@glsfirst ... 144, 458, 709, 962, 963
\@glsfirst@ 144, 358, 458, 557, 709, 810, 963
\@glsfirstletter 179, 180, 473, 725, 979, 980
\@glsfirstplural . 146, 458, 710, 963
\@glsfirstplural@ 146, 358, 358, 458, 557, 710, 810, 963
\@glshypernumber 238, 238, 510, 761, 762, 1016
\@glsisacronymlistfalse 22, 385, 634, 889
\@glsisacronymlisttrue 22, 385, 634, 889
\@glslink 104, 124, 136, 136, 177, 271, 446, 453, 454, 472, 596, 697, 705, 706, 724, 849, 939, 940, 950, 959, 979, 1033
\@glslocalreset 105, 106, 109, 433, 434, 683–685, 940–942
\@glslocalunset 106, 106, 109, 433, 434, 684, 685, 941, 942
\@glslocref 200, 205, 206, 208, 209, 488, 490–492, 576, 740, 742–744, 829, 995, 998, 999
\@glsminrange 180, 180, 182, 474, 475, 577, 578, 725–727, 830, 831, 980, 981
\@glsname 147, 459, 710, 964
\@glsname@ 147, 359, 459, 557, 710, 811, 964
\@glsnavhypertarget 273, 273, 596, 849, 1033
\@glsnextpages 215, 229, 495, 505, 747, 756, 1003, 1012
\@glsnodesc 80, 95, 97, 99, 417, 427, 430, 668, 677, 678, 680, 924, 933, 934, 936, 937
\@glsnoname 80, 95, 97, 99, 417, 427–429, 667, 677, 679, 680, 923, 933, 935, 936
\@glsnonextpages 215, 229, 495, 504, 747, 756, 1003, 1012
\@glsnumberformat 120, 123, 177, 200, 209, 444, 445, 472, 488, 492, 576, 695, 696, 724, 740, 743, 744, 829, 948, 950, 979, 995, 999
\@glsopenfile 190, 191, 201, 481, 489, 733, 740, 988, 996
\@glsorder 32, 194, 195, 392, 484, 641, 736, 896, 991
\@glspl 139, 455, 707, 960
\@glspl@ 109, 112, 139, 139, 333–335, 435, 436, 455, 593–595, 686, 687, 707, 847, 848, 943, 944, 960
\@glsplural 145, 458, 710, 963
\@glsplural@ 145, 358, 458, 557, 710, 810, 963
\@glsprefix@record@hook .. 332, 332, 333–336, 593–595, 846–849
\@glsreset 105, 107, 109, 432–434, 683–685, 940–942
\@glssee 100, 211, 430, 493, 681, 745, 937, 1001
\@glsshowaccsupp . 7, 9, 338, 377, 378, 540, 627, 628, 793, 881, 882
\@glsshowtarget 7, 8, 8, 377, 378, 453, 626, 627, 705, 880–882
\@glssymbol 150, 460, 711, 965
\@glssymbol@ 150, 360, 460, 558, 711, 811, 965
\@glssymbolplural 151, 460, 712, 965
\@glssymbolplural@ 151, 360, 460, 558, 712, 811, 965

\@glstarget 136, 136, 136, 232, 273, 454, 506, 596, 705, 706, 758, 849, 959, 1013, 1033
\@glstext 143, 457, 709, 962
\@glstext@ 143, 357, 457, 557, 709, 810, 962
\@glsunset 106, 106, 109, 433, 434, 684, 685, 941, 942
\@glsuseri 152, 460, 712, 966
\@glsuseri@ 152, 360, 460, 559, 712, 812, 966
\@glsuserii 153, 461, 712, 966
\@glsuserii@ 153, 361, 461, 559, 712, 812, 966
\@glsuseriii . . 154, 461, 713, 966, 967
\@glsuseriii@ 154, 361, 461, 559, 713, 812, 967
\@glsuseriv 155, 461, 713, 967
\@glsuseriv@ 155, 361, 462, 559, 713, 812, 967
\@glsuserv 156, 462, 713, 967
\@glsuserv@ 156, 362, 462, 559, 714, 813, 967
\@glsuservi 157, 462, 714, 968
\@glsuservi@ 157, 362, 462, 560, 714, 813, 968
\@glswidestname 324, 324, 325, 326, 586, 587, 624, 625, 840, 877, 878
\@glswritefiles 38, 201, 394, 488, 644, 740, 899, 995
\@glsxtr@doaccsupp 337, 539, 793
\@glsxtr@record 332, 593, 846
\@gobble 7, 16–18, 34–36, 86, 104, 127, 179, 183, 195, 197, 278, 377, 382, 383, 392, 393, 420, 421, 432, 447, 473, 475, 476, 484, 485, 575, 578, 579, 626, 627, 631–633, 642, 643, 671, 672, 683, 698, 724, 725, 727, 736, 737, 828, 831, 832, 852, 880, 881, 886, 887, 897, 898, 927, 928, 939, 952, 979, 982, 991, 992
\@gobblethree 7, 377, 627, 881
\@idxitem 319, 620, 874
\@ifclassloaded 6, 14, 51, 376, 380, 401, 626, 630, 651, 880, 884, 907
\@ifglossaryexists . . 64, 64, 213, 409, 494, 659, 660, 746, 915, 1001
\@ifl@t@r 27, 388, 638, 892
\@ifnextchar 73, 121, 414, 444, 664, 695, 920, 949
\@ifpackageloaded 6, 11, 12, 29–31, 46, 62, 103, 175, 187, 336, 376, 379, 390, 391, 398, 408, 432, 471, 478, 539, 626, 629, 639–641, 648, 659, 683, 723, 730, 792, 880, 883, 884, 894, 895, 904, 914, 939, 978, 985
\@ifstar 64, 72, 87, 88, 121, 179, 180, 241, 409, 413, 422, 444, 473, 511, 659, 663, 672, 673, 695, 725, 763, 915, 919, 928, 929, 949, 979, 980, 1017
\@ifundefined 45, 282, 293, 305, 312, 326, 398, 587, 596, 600, 606, 613, 617, 624, 625, 648, 840, 849, 854, 860, 866, 867, 870, 871, 878, 904
\@ignored@glossaries 35, 74, 75, 75, 393, 414, 415, 643, 664, 665, 898, 920, 921
\@input@ 216, 496, 747, 1003
\@istfilename 48, 194, 195, 400, 484, 650, 736, 906, 991
\@makecol 292, 605, 606, 859
\@makeglossaries@warn@noprintglossary 195, 196, 736, 737, 991, 992
\@makeglossary 34, 190, 195, 392, 393, 481, 484, 642, 643, 733, 736, 897, 988, 991
\@minus 278, 299, 318, 599, 609, 620, 852, 863, 874
\@mkboth . . 52, 401, 402, 651, 652, 907
\@ne 104, 939
\@newglossary 71, 73, 73, 413, 414, 663, 664, 919, 920
\@newglossaryentry@defcounters 100, 101, 108, 430, 431, 434, 681, 685, 937, 942
\@newglossaryentryposthook 88, 89, 100, 101, 330, 341, 422, 430, 431, 542, 592, 673, 681, 796, 845, 929, 937
\@newglossaryentryprehook 88, 89, 94, 96, 101, 330, 340, 341, 422, 426, 427, 431, 542, 591, 673, 677, 678, 681, 795, 845, 929, 933, 934, 937

\@nil 21, 100, 126–128, 168, 188, 209, 211, 224, 225, 238–240, 385, 430, 447, 448, 468, 479, 480, 492, 493, 501, 502, 510, 576, 634, 681, 698, 699, 719, 731, 732, 743, 745, 753, 754, 761, 762, 829, 889, 937, 951–953, 974, 986, 999, 1000, 1008, 1009, 1016, 1017
\@nnil 21, 211, 385, 493, 634, 745, 889, 1001
\@no@makeglossaries 34, 195, 198, 200, 392, 393, 485, 486, 488, 642, 736, 738, 740, 897, 992, 993, 995
\@no@post@desc 47, 400, 581, 650, 834, 905
\@nopostdesc 47, 215, 399, 495, 650, 747, 905, 1003
\@onelevel@sanitize 8, 25, 35, 58, 86, 102, 127, 180, 184, 210, 212, 224, 240, 378, 387, 406, 420, 432, 447, 448, 473, 474, 476, 493, 494, 501, 577, 578, 627, 636, 637, 643, 656, 671, 682, 698, 725, 728, 744, 746, 753, 830, 832, 882, 891, 898, 911, 927, 938, 952, 980, 982, 1000, 1001, 1008, 1017
\@onlypreamble . . . 73, 83, 94, 95, 110, 113, 196, 199, 414, 420, 426, 435, 437, 485, 487, 664, 670, 677, 686, 688, 737, 739, 920, 926, 933, 943, 945, 992, 995
\@onlypremakeg 43, 48, 49, 55, 57, 60, 73, 186, 397, 400, 404, 407, 414, 478, 647, 650, 654, 655, 657, 664, 730, 903, 906, 909, 910, 913, 920, 985
\@org@Gls@acrentryname . . . 168, 974
\@org@glossaryentrynumbers 215, 216, 495, 747, 1002, 1003
\@org@gls@assign@descplural 524, 531, 533, 535, 569, 572, 573, 776, 783, 784, 786, 788, 823, 825, 826
\@org@gls@assign@firstpl 524–527, 529, 531, 533–535, 569–573, 775, 776, 778–784, 786, 788, 822–826
\@org@gls@assign@plural 524–527, 529, 531, 533–535, 569–573, 776, 778–784, 786, 788, 823–826
\@org@gls@assign@symbolplural 524–527, 529, 533–535, 570, 571, 573, 776, 778–782, 786, 788, 823, 824, 826
\@org@glsnumberformat 176, 471, 723, 978
\@org@newglossaryentryprehook 94, 426, 677, 933
\@outputpage 292, 605, 606, 859
\@p@glossarysection 51, 53, 401, 402, 651, 652, 907, 908
\@pgls 332, 332, 593, 846
\@pgls@ 332, 332, 593, 846
\@pglspl 333, 333, 593, 846
\@pglspl@ 333, 333, 593, 846
\@plus 278, 299, 318, 599, 609, 620, 852, 863, 874
\@print@glossary 213, 216, 494, 496, 746, 747, 1002, 1003
\@print@noidx@glossary . . . 214, 223, 494, 500, 746, 752, 1002, 1008
\@printgloss@checkexists 213, 213, 214, 215, 494, 495, 746, 747, 1002
\@printgloss@checkexists@allowignored 213, 494, 746, 1002
\@printgloss@checkexists@noignored 213, 213, 214, 494, 746, 1001, 1002
\@printgloss@setsort . . 195, 198, 214, 215, 485, 486, 494, 495, 736, 738, 746, 747, 992, 993, 1002
\@printglossary 213, 214, 214, 494, 746, 1002
\@roman . . . 58, 405, 577, 656, 830, 911
\@secondofthree 121, 136, 138, 140, 159, 161, 163, 165, 355, 444, 454–456, 463–465, 467, 555, 695, 706, 708, 715–718, 808, 949, 959–961, 969–972
\@secondoftwo 26, 29–31, 46, 86, 87, 104, 136–139, 142, 158, 159, 162, 163, 205, 278, 328, 355–357, 387, 388, 390, 391, 398, 420, 421, 454, 455, 457, 463, 465, 466, 555, 556, 574, 591, 637,

639–641, 648, 671, 672, 705–708, 714, 715, 717, 808–810, 827, 844, 852, 891, 892, 894, 895, 904, 927, 928, 939, 940, 959, 960, 962, 968, 969, 971, 998
\@set@glo@numformat 126, 209, 447, 492, 576, 698, 744, 829, 951, 999
\@sglsaddkey 88, 422, 673, 929
\@sglsaddstoragekey 87, 422, 672, 928
\@tabacckludge 26, 387, 637, 891
\@text@composite@x . 26, 387, 637, 891
\@thirdofthree . . . 121, 139, 141, 159, 161, 163, 165, 356, 444, 455, 456, 463, 465–467, 555, 695, 707, 708, 715–717, 719, 809, 949, 960, 961, 969–971, 973
\@this@attr . . . 183, 475, 476, 727, 982
\@this@childlabel 218, 219, 497, 749, 1004
\@this@counter 56, 404, 654, 910
\@this@ctr 183, 475, 727, 982
\@this@key 87, 421, 672, 928
\@this@label . 218, 497, 748, 749, 1004
\@thiscs 43, 397, 398, 648, 903
\@tmp 406, 577, 830
\@use@ption 41, 396, 646, 902
\@warn@nomakeglossaries . . 193, 217, 484, 496, 735, 748, 990, 1004
\@wrglossary@pageformat 204, 205, 490, 742, 997
\@wrglossarynumberhook 204, 205, 208, 490, 491, 742, 743, 997, 999
\@write@glslabelsname 35, 36, 643, 898
\@xdy@main@language 33, 33, 392, 482, 483, 496, 642, 734, 735, 748, 896
\@xdyattributelist 54, 56, 183, 403, 404, 475, 653, 654, 727, 909, 910, 982
\@xdyattributes 54, 56, 181, 403, 404, 474, 575, 577, 653, 654, 726, 828, 830, 909, 910, 981
\@xdycounters 55, 56, 183, 403, 404, 475, 653, 654, 727, 909, 910, 982
\@xdycrossrefhook 59, 183, 406, 475, 657, 727, 912, 982
\@xdylanguage 62, 217, 408, 496, 658, 748, 914, 1003
\@xdylettergroups 63, 63, 184, 409, 477, 579, 659, 728, 832, 914, 915, 983
\@xdylocationclassorder 60, 61, 183, 407, 475, 578, 657, 727, 831, 913, 982
\@xdylocref 55, 56, 184, 403, 404, 476, 575, 579, 653, 654, 728, 828, 832, 909, 910, 983
\@xdynumbergrouporder 63, 180, 180, 409, 473, 474, 659, 725, 915, 980
\@xdyrequiredstyles 61, 61, 181, 408, 474, 576, 658, 726, 830, 914, 980
\@xdysortrules . . 61, 61, 184, 407, 477, 579, 658, 728, 832, 913, 983
\@xdystyle 181, 474, 576, 577, 726, 830, 980, 981
\@xdyuseralphabets 57, 58, 181, 405, 474, 577, 655, 726, 830, 911, 981
\@xdyuserlocationdefs 59, 60, 182, 406, 407, 475, 575, 578, 656, 657, 727, 829, 831, 912, 913, 981
\@xdyuserlocationnames 59, 60, 406, 407, 576, 656, 657, 829, 912, 913
\@xfor@nextelement 211, 493, 745, 1001
\@xp@gls@getcounterprefix 206, 208, 209, 998, 999
\@xp@glshypernumber . . 238, 238, 1016
\\ 101, 127, 132, 133, 179, 185, 186, 238, 239, 431, 447, 473, 477, 478, 510, 579, 580, 582, 583, 589, 590, 682, 698, 703, 704, 725, 729, 762, 832, 833, 835, 836, 842, 843, 938, 952, 957, 979, 983, 984, 1016, 1017
\^ 26, 35, 387, 637, 643, 891, 898
\‘ 26, 387, 637, 891
\| 128, 130, 132, 188, 448, 450, 480, 699, 701, 703, 731, 953–955, 957, 986
\~ 26, 388, 637, 891

A

\A 235, 1014
\a 26, 388, 637, 892
\AA 27, 388, 637, 892
\aa 27, 388, 637, 892

\Ac 260, 523, 775, 1030
\ac 260, 523, 775, 1030
access (key) 338
\accsuppglossaryentryfield 363, 540, 561, 793, 814
\accsuppglossarysubentryfield 363, 540, 561, 793, 814
\Acf 260, 523, 775, 1030
\acf 260, 523, 775, 1030
\Acfp 260, 523, 775, 1030
\acfp 260, 523, 775, 1030
\Acl 260, 523, 775, 1030
\acl 260, 523, 775, 1030
\Aclp 260, 523, 775, 1030
\aclp 260, 523, 775, 1030
\Acp 261, 523, 775, 1030
\acp 260, 523, 775, 1030
\acrfootnote 261, 524, 525, 530, 777, 783, 1030
\ACRfull 245, 512, 764, 1019
\Acrfull 244, 260, 512, 523, 763, 775, 1019, 1030
\acrfull 244, 244, 260, 511, 523, 763, 775, 1018, 1019, 1030
\ACRfullfmt 244, 245, 247, 256, 257, 368, 370, 512, 514, 520, 521, 565, 567, 764, 765, 772, 773, 818, 820, 1018–1020, 1026, 1028
\Acrfullfmt 243, 245, 247, 255, 257, 368, 370, 512, 513, 520, 521, 565, 567, 764, 765, 772, 773, 818, 820, 1018–1020, 1026, 1028
\acrfullfmt 243, 244, 247, 255, 257, 368, 370, 511, 513, 520, 521, 565, 566, 763, 765, 771, 773, 818, 820, 1018–1020, 1026, 1028
\acrfullformat 175, 243, 243, 471, 511, 512, 523, 536, 723, 763, 775, 789, 977, 1018
\ACRfullpl 246, 512, 764, 1019
\Acrfullpl 245, 260, 512, 523, 764, 775, 1019, 1030
\acrfullpl 245, 245, 260, 512, 523, 764, 775, 1019, 1030
\ACRfullplfmt 244, 246, 247, 256, 257, 369, 371, 513, 514, 520, 521, 565, 567, 764–766, 772, 773, 818, 820, 1018–1020, 1027, 1028
\Acrfullplfmt 244, 245, 247, 256, 257, 369, 371, 512, 514, 520, 521, 565, 567, 764, 766, 772, 773, 818, 820, 1018–1020, 1027, 1028
\acrfullplfmt 244, 245, 247, 256, 257, 368, 371, 512, 514, 520, 521, 565, 567, 764, 766, 772, 773, 818, 820, 1018–1020, 1026, 1028
\acrlinkfootnote 261, 261, 524, 777, 1030
\acrlinkfullformat 243, 243, 244, 511–513, 763–765, 1018
\ACRlong 163, 466, 717, 971
\Acrlong 162, 162, 260, 465, 523, 717, 775, 971, 1030
\acrlong 161, 260, 465, 523, 716, 775, 970, 1030
\ACRlongpl 165, 467, 718, 972
\Acrlongpl 164, 164, 260, 466, 523, 718, 775, 972, 1030
\acrlongpl 164, 260, 466, 523, 718, 775, 972, 1030
\acrnameformat 246, 513, 528, 571, 765, 781, 824, 1019
\acrnolinkfootnote 262, 524, 777, 1030
acronym (option) 20
\acronymentry 247, 248, 250–254, 256–259, 364–367, 369–373, 513–523, 561–568, 765–775, 815–817, 819–822, 1020–1025, 1027–1029
\acronymfont 116, 117, 158–161, 168, 175, 246, 246, 248, 250–259, 352, 353, 355–357, 364–373, 441, 442, 463–465, 468, 471, 513–523, 525, 526, 528–534, 553–556, 561–568, 570–572, 692, 693, 714–716, 720, 723, 765–775, 777, 779, 780, 782–785, 787, 806–810, 815–823, 825, 946, 947, 968–970, 974, 977, 1019, 1021–1029
acronymlists (option) 22
\acronymname 20, 44, 47, 383, 384, 398, 399, 633, 634, 648, 649, 888, 903, 905
acronyms (option) 20

\acronymsort . . 247, 248, 250–254, 256–259, 364–367, 369, 370, 372, 373, 513–523, 561–563, 566, 568, 765–775, 815–817, 819–822, 1020–1025, 1027–1029
\acronymtype 19, 20, 247, 248, 383, 384, 513, 514, 523–537, 569–572, 633, 634, 765, 766, 775–790, 822–825, 887, 888, 1020, 1021
\acrpluralsuffix 242, 247, 250–252, 256–258, 364, 365, 369–372, 511, 513, 515–517, 520–527, 529–531, 533, 534, 536, 561, 562, 566, 568–573, 763, 765, 767–769, 772–775, 777–779, 781–784, 786, 787, 789, 790, 815, 819–826, 1018, 1020, 1022–1024, 1027–1029
\ACRshort 159, 463, 715, 969
\Acrshort 158, 158, 259, 463, 523, 715, 775, 968, 1029
\acrshort 158, 259, 463, 523, 714, 775, 968, 1029
\ACRshortpl 161, 464, 716, 970
\Acrshortpl 160, 160, 259, 464, 523, 716, 775, 970, 1029
\acrshortpl 160, 259, 464, 523, 715, 775, 969, 1029
acrstyles:
dua 254, 367
dua-desc 256, 369
footnote 256, 370
footnote-desc 258, 372
footnote-sc 258, 371
footnote-sc-desc 259, 372
footnote-sm 258, 372
footnote-sm-desc 259, 372
long-sc-short 251
long-sc-short-desc 253, 365
long-short 249, 364
long-short-desc 252, 365
long-sm-short 252
long-sm-short-desc 253, 366
long-sp-short 250
long-sp-short-desc 253
sc-short-long 252
sc-short-long-desc 254, 366
short-long 251, 364
short-long-desc 253, 366
sm-short-long 252
sm-short-long-desc 254, 367
\Acs 259, 523, 775, 1029
\acs 259, 523, 775, 1029
\Acsp 259, 523, 775, 1029
\acsp 259, 523, 775, 1029
\add@accent@ 26, 387, 637, 891
\addcontentsline . . 54, 403, 653, 909
\addglossarytocaptions 45, 46, 398, 648, 649, 904
\addtolength . . 326, 587, 625, 840, 878
\advance 17, 18, 97, 104, 125, 292, 382, 383, 428, 446, 605, 632, 679, 697, 859, 886, 887, 935, 939, 951
\AE 26, 388, 637, 892
\ae 26, 388, 637, 892
altlist (style) 280
altlistgroup (style) 280
altlisthypergroup (style) 280
altlong4col (style) 287
altlong4col-booktabs (style) 290
altlong4colborder (style) 287
altlong4colheader (style) 287
altlong4colheaderborder (style) . 288
altlongragged4col (style) 296
altlongragged4col-booktabs (style) 291
altlongragged4colborder (style) . 298
altlongragged4colheader (style) . 297
altlongragged4colheaderborder (style) 298
\altnewglossary . . . 73, 414, 664, 920
altsuper4col (style) 310
altsuper4colborder (style) 311
altsuper4colheader (style) 310
altsuper4colheaderborder (style) 311
altsuperragged4col (style) 316
altsuperragged4colborder (style) 317
altsuperragged4colheader (style) 317
altsuperragged4colheaderborder (style) 317
alttree (style) 324
alttreegroup (style) 327
alttreehypergroup (style) 327
\andname 45, 212, 398, 493, 648, 745, 904, 1001
\AnyTrackedLanguages 46, 328, 329, 399, 574, 591, 649, 827, 844, 905
\appto 22, 27, 79, 88, 89, 205, 330, 340,

385, 388, 417, 422, 542, 591, 592, 635, 638, 667, 673, 795, 844, 845, 889, 892, 923, 929, 998
\AtBeginDocument 20, 62, 85, 103, 178, 197, 384, 408, 420, 432, 472, 486, 633, 659, 671, 683, 724, 738, 888, 914, 927, 939, 979, 993
\AtEndDocument 35, 37, 85, 110, 196, 201, 217, 274, 393, 394, 420, 435, 485, 488, 496, 596, 643, 644, 671, 686, 737, 740, 748, 849, 898, 899, 927, 943, 992, 995, 1003, 1033
automake (option) 36
automakegloss (option) 37
automakeglosslite (option) 37

B

\b 26, 388, 637, 892
\begin 183, 223, 279, 282–288, 290, 291, 293–318, 476, 500, 578, 599–603, 605–620, 727, 752, 831, 852, 854–874, 982, 1008
\BeginAccSupp 337, 540, 793
\begingroup 6, 8, 35, 202, 207, 216, 239, 376, 378, 489, 491, 510, 626, 627, 643, 741, 743, 762, 880, 882, 898, 996, 998, 1016
\bfseries 283, 285–287, 289, 290, 294–296, 298, 306, 308–311, 313–318, 601–604, 607–609, 614–620, 854–858, 860–863, 867–874
\bgroup 26, 94, 132, 176, 209, 218, 239, 387, 426, 471, 495, 497, 637, 677, 703, 723, 747, 749, 891, 933, 957, 978, 999, 1002, 1004, 1017
\bool 135
\boolean 536, 789
\boolfalse 38, 394, 644, 900
\booltrue 394, 644
\bottomrule . . . 289, 290, 604, 857, 858
\box 292, 605, 606, 859

C

\c 26, 388, 637, 892
\c@equation 125, 446, 697, 951
\c@glossaryentry . . 14, 381, 630, 884
\c@glossarysubentry 15, 381, 631, 885
\c@page 204, 205, 208, 490, 491, 742, 743, 997–999
\capitalisewords 171, 721, 975
\catcode 35, 85, 132, 420, 643, 671, 703, 898, 927, 957
\cGls 111, 111, 112, 436, 687, 944
\cgls 111, 111, 436, 687, 944
\cGlsformat 110, 112, 435, 436, 686, 687, 943, 944
\cglsformat 109, 111, 435, 436, 686, 687, 943, 944
\cGlspl . . 112, 113, 436, 437, 687, 944
\cglspl . . 112, 112, 113, 436, 687, 944
\cGlsplformat 110, 113, 435, 437, 686, 688, 943, 945
\cglsplformat 109, 112, 435, 436, 686, 687, 943, 944
\changes 175
\char 235, 508, 760, 1015
\cleardoublepage 53, 402, 652, 653, 908
\clearpage . . . 53, 402, 652, 653, 908
\closeout . . 35, 85, 185, 186, 191, 201, 393, 420, 477, 478, 481, 489, 643, 671, 728, 730, 733, 741, 898, 927, 983, 984, 988, 996
\compatglossarystyle 580–590, 834–844
compatible-2.07 (option) 38
compatible-3.07 (option) 38
\compatibleglossentry 232, 337, 506, 507, 540, 758, 759, 793
\compatiblesubglossentry 234, 337, 507, 540, 759, 793
\copy 292, 605, 606, 859
\count@ 224, 501, 753, 1008, 1009
counter (key) 77
counter (option) 22
counters:
  glossaryentry 14
  glossarysubentry 15
counterwithin (option) 14
\cs 272, 273
\csdef 24, 88–90, 100, 101, 108, 109, 218, 219, 237, 249, 386, 422–424, 430, 431, 434, 435, 497, 509, 514, 515, 580, 636, 673, 674, 681, 682, 685, 748, 749, 761,

766, 834, 890, 929, 930, 937, 938, 942, 1004, 1005, 1016, 1021
\csedef 110, 204, 435, 490, 686, 742, 943, 997
\csgdef 19, 51, 71, 74, 109, 110, 212, 227, 401, 413, 414, 434, 435, 494, 503, 651, 663, 664, 685, 686, 746, 755, 887, 907, 919, 920, 942, 943, 1001, 1010
\cslet 78, 79, 94, 101, 222, 417, 426, 431, 500, 667, 677, 682, 752, 922, 923, 933, 938, 1007
\csname 15–18, 26, 41, 44, 46, 47, 52, 53, 56, 58, 59, 62, 64, 67, 72–74, 81, 82, 88–90, 92, 93, 97, 98, 100–104, 106, 107, 123–126, 137–142, 158–165, 176, 177, 182, 183, 188–191, 197, 200, 201, 203, 204, 209–211, 214–217, 219, 237, 241, 265–270, 324, 326, 330, 341, 342, 345, 349, 355–357, 374, 375, 381–383, 387, 396, 398, 399, 402, 404, 406, 408–410, 413, 414, 418, 422–425, 428–433, 443, 445–447, 454–457, 463–467, 472, 474–476, 479–481, 486, 488–490, 492, 493, 495–497, 503, 506, 507, 509, 511, 537–540, 542–545, 548, 555, 556, 573–576, 587, 592, 596, 597, 624, 625, 631–633, 637, 646, 648, 649, 652, 654, 656, 658, 659, 661, 663, 664, 668, 669, 672–676, 679–684, 694, 696–698, 706–708, 714–719, 723, 724, 726, 727, 731–733, 737, 740–742, 744–749, 755, 758–761, 763, 776, 790–793, 796–798, 801, 808–810, 826–829, 840, 845, 849, 850, 877, 878, 885–887, 891, 902–905, 908, 910, 912, 914–916, 919, 920, 924, 925, 928–932, 935–939, 941, 950, 951, 960–962, 968–973, 978, 979, 981, 982, 986–988, 993, 995–997, 999, 1000, 1002, 1003, 1005, 1015, 1017, 1031–1033
\csshow 269, 539, 792, 1032
\csuse 47, 50, 71, 78, 81, 82, 89, 90, 166, 191–193, 219, 221, 223, 225, 227–229, 249, 330, 399, 401, 413, 418, 419, 423, 424, 443, 444, 482, 483, 497, 499–501, 503, 504, 509, 515, 581–590, 592, 649, 651, 663, 668, 669, 674, 694, 734, 735, 749, 751–753, 755, 756, 761, 767, 835–839, 841–845, 905, 906, 919, 922, 924, 925, 930, 973, 989, 990, 1005, 1006, 1008–1011, 1021
\csxdef 99, 110, 429, 435, 680, 686, 776, 936, 943
\currentglossary 14, 15, 50, 215, 271, 273, 274, 381, 401, 495, 630, 631, 651, 747, 885, 906, 1003, 1033
\currentglssubentry 15, 231, 381, 505, 631, 757, 885, 1012
\CurrentOption 41, 329, 336, 396, 539, 591, 646, 792, 844, 902
\CurrentTrackedLanguage 46, 328, 329, 399, 574, 591, 649, 828, 844, 905
\CurrentTrackedTag . 46, 328, 329, 399, 574, 591, 649, 827, 844, 905
\CustomAcronymFields 264, 536, 789, 790
\CustomNewAcronymDef 264, 536, 537, 789, 790

D

\d 26, 387, 637, 891
\day 181, 185, 474, 477, 576, 579, 726, 729, 830, 832, 980, 983
debug (option) 7
\DeclareAcronymList 20, 21, 22, 247, 248, 383–385, 513, 514, 524, 526, 527, 529, 531, 533, 535, 537, 633–635, 765, 766, 776, 778, 780, 782, 784, 786, 788, 790, 888, 889, 1020, 1021
\DeclareCurrentRelease 4, 271, 275, 277, 281, 288, 293, 299, 304, 312, 318, 328, 329, 336, 375, 376, 626, 879
\DeclareListParser 195, 485, 736, 991

\DeclareOption 11, 329, 336, 379, 539, 591, 629, 792, 844, 883
\DeclareOptionX ... 11, 379, 629, 883
\DeclareRelease 4, 271, 275, 277, 281, 288, 293, 299, 304, 312, 318, 328, 329, 336, 375, 376, 626, 879
\DeclareRobustCommand 47, 399, 532, 650, 784, 905
\def 11, 12, 14–21, 25–27, 32, 33, 35, 37–40, 42–44, 47, 48, 51, 55, 57–63, 67, 71–80, 82, 92, 94–99, 101, 104, 105, 109–113, 120, 123, 125–134, 137–165, 176, 177, 181, 185, 187–189, 191–194, 196–199, 201, 204, 205, 208–211, 214, 215, 218, 222–224, 226–229, 235–241, 244–247, 274–277, 303, 304, 324–327, 329, 330, 332–336, 338–341, 355–362, 379–384, 387, 388, 392, 394–401, 403, 405–411, 413–417, 419, 425–432, 435–437, 443–467, 472, 474, 477–484, 486–488, 490–493, 495, 496, 500–504, 508–513, 524–529, 531–533, 535, 537, 540–542, 555–560, 569–573, 576, 579, 581, 586, 587, 591, 593–598, 612, 613, 624, 625, 629–634, 636–638, 642–645, 647–651, 653, 655–661, 663–667, 669, 675, 677–683, 685–688, 694–719, 723, 724, 726, 728, 730–735, 738–740, 742–748, 752–756, 760–765, 776, 778–786, 788, 790, 793–796, 808–813, 823–826, 829, 830, 832, 834, 840, 844–848, 850, 851, 865, 866, 877–879, 883–889, 891, 892, 896, 898–903, 905–907, 909, 911–916, 919–923, 925, 932–940, 942–945, 948, 950–973, 978–980, 983, 985–995, 997–1002, 1004, 1007, 1008, 1010–1012, 1014–1020, 1033
\def@gls@xdycheckbackslash 133, 134, 452, 453, 704, 705, 958
\DefaultNewAcronymDef 261, 373, 523, 524, 569, 775, 777, 822
\defglsdisplay 119, 443, 694
\defglsdisplayfirst ... 119, 443, 694
\defglsentryfmt 71, 73, 74, 248, 261, 413, 414, 443, 514, 523, 524, 526, 528, 530, 532, 534, 536, 663–665, 694, 766, 775, 777, 779, 780, 782, 785, 787, 789, 919, 920, 1021, 1030
\define@boolkey 10, 12–15, 20, 27, 28, 31–33, 38, 120, 121, 229, 379–381, 383, 388, 389, 391, 392, 394, 444, 504, 628–631, 633, 638, 641, 642, 644, 695, 756, 883–885, 888, 892, 893, 895, 896, 899, 900, 948, 1011
\define@choicekey 7, 9, 10, 15, 28, 30, 32, 36, 39, 41, 78, 228, 229, 377–379, 381, 389, 390, 392, 393, 395, 416, 503, 504, 626, 628, 629, 631, 639–641, 644, 645, 666, 755, 756, 880, 882, 883, 885, 893, 894, 896, 898, 900, 901, 922, 1011
\define@key 11, 14, 22, 23, 29, 31, 33, 75–80, 88, 89, 119, 120, 177, 227, 229, 329, 338–340, 379, 381, 385, 389, 392, 415–417, 422, 444, 472, 503, 504, 540, 541, 591, 629, 631, 635, 639, 642, 665–667, 673, 695, 724, 755, 756, 794, 795, 844, 883, 885, 889, 893, 895, 896, 921–923, 929, 948, 979, 1010–1012
\DefineAcronymSynonyms 42, 259, 261, 397, 523, 647, 775, 902, 1029, 1030
\delimN 50, 184, 195, 226, 239, 401, 476, 485, 502, 510, 578, 651, 728, 736, 754, 762, 832, 906, 982, 992, 1010, 1016
\delimR 50, 184, 238, 239, 401, 476, 510, 578, 651, 728, 762, 832, 906, 982, 1016
description (key) 75
description (option) 32
descriptionaccess (key) 339
\DescriptionDUANewAcronymDef 262, 526, 528, 779, 780

\DescriptionFootnoteNewAcronymDef 262, 373, 525, 526, 570, 777, 778, 823
\descriptionname 44, 47, 283, 285–287, 289, 290, 294–296, 298, 306, 308–311, 313–318, 398, 399, 601–604, 607–609, 614–616, 618–620, 648, 649, 854–858, 860–863, 867–874, 903, 905
\DescriptionNewAcronymDef . 262, 373, 528, 529, 570, 781, 782, 824
descriptionplural (key) 75
descriptionpluralaccess (key) . . 339
\DH 27, 388, 638, 892
\dh 27, 388, 638, 892
\dimen@ 251, 292, 324, 516, 605, 623, 624, 768, 859, 877, 1023
\directlua 42, 902
\disable@keys 42, 396, 646, 902
disablemakegloss (option) 34
\do 31, 41, 43, 45, 55, 56, 63, 64, 86, 87, 127, 176, 181–183, 194–196, 203, 211, 218, 248, 391, 396, 397, 403, 404, 409, 420, 421, 447, 472, 474, 475, 484, 485, 490, 493, 497, 514, 524, 526, 528, 529, 531, 534, 535, 537, 576, 596, 597, 641, 646, 648, 654, 659, 671, 672, 698, 723, 726, 727, 736, 737, 742, 745, 748, 749, 766, 776, 779, 780, 782, 784, 787, 788, 790, 830, 849, 850, 895, 902–904, 909, 910, 915, 927, 928, 952, 978, 980–982, 990–992, 997, 1001, 1004, 1021, 1033
\do@glo@storeentry 16–19, 100, 381–383, 430, 631–633, 681, 885–887, 937
\do@gls@link@checkfirsthyper 122, 124, 137–143, 158–165, 355–357, 445, 454–457, 463–467, 555, 556, 695, 696, 706–709, 714–718, 808, 809, 949, 950, 959–962, 968–973
\do@gls@xdycheckbackslash 127, 447, 698, 952
\do@glsdisablehyperinlist 123, 124, 445, 696, 950
\do@glshaschildren 67, 410, 411, 660, 661, 916
\do@write@glslabelsname 35, 35, 36, 643, 898
\doifglossarynoexistsordo . . 66, 72, 410, 413, 660, 663, 916, 919
\dtl@ifsingle 508, 760
\dtl@insertinto . 219, 498, 750, 1005
\dtl@sortresult 220, 498, 750, 1005, 1006
\dtlcompare 220, 498, 750, 1006
\dtlicompare 220, 498, 750, 1006
\DTLifinlist 75, 123, 415, 445, 665, 696, 921, 950
\DTLifint 235, 508, 760, 1015
\dtlletterindexcompare 220, 498, 750, 1005
\DTLsubstituteall 127, 448, 698, 952
\dtlwordindexcompare 220, 498, 750, 1005
dua (acrstyle) 254, 367
dua (option) 32
dua-desc (acrstyle) 256, 369
\DUANewAcronymDef 263, 534, 535, 787, 788

E

\eappto 414, 415, 431, 490
\edef . . . 43, 46, 54–56, 58, 60, 61, 72, 81, 82, 85, 87, 125, 127–134, 179, 185, 187, 189, 191–193, 195, 201, 217, 220, 224, 235, 239–241, 383, 384, 397, 399, 403–410, 413–415, 418–421, 424–427, 431, 443, 445–453, 471–473, 477–480, 482, 483, 485, 488, 491, 492, 494, 496–498, 501, 505, 508, 510, 511, 523, 525, 526, 528, 530, 532, 534, 569–572, 575–577, 579, 596, 648, 649, 653–658, 663, 664, 668, 669, 671, 672, 694, 697–705, 724, 725, 728, 730–732, 734–736, 740, 743, 744, 748, 750, 753, 760, 762, 763, 775, 777, 779, 781, 783, 785, 787, 822–825, 828–830, 832, 903, 904, 909–914, 919, 924, 925, 927, 928, 950–958,

979, 983, 985–991, 995, 1003, 1005, 1006, 1009, 1015, 1017
\egroup 26, 94, 176, 209, 219, 240, 387, 426, 472, 495, 497, 637, 677, 724, 747, 749, 891, 933, 978, 1000, 1003, 1004, 1017
\else 8, 13–15, 18–21, 23, 25, 27, 28, 35, 37, 38, 41–43, 48, 49, 51–58, 60–63, 71, 78, 81, 97, 98, 101–104, 109, 110, 120, 123–134, 168, 169, 180, 181, 184–187, 189, 190, 192–194, 201, 203–205, 208–211, 215, 224, 229, 231, 236, 238–240, 251, 263, 265, 279, 283, 284, 286, 289, 290, 292, 294, 295, 297, 306, 307, 309, 313, 315, 317, 320, 322, 323, 325, 326, 346, 377, 378, 380, 381, 383–386, 388, 389, 393–397, 400–409, 416, 418, 428, 429, 431, 432, 435, 445–457, 468, 473, 474, 476–483, 488–493, 495, 501, 504–506, 508, 510, 516, 526, 527, 529, 531, 532, 534–537, 545, 576–581, 585–587, 596, 599, 601, 602, 604–608, 613–615, 617–619, 621–625, 627, 630–636, 638, 639, 643–648, 650–659, 666, 668, 679, 680, 682, 683, 686, 696–708, 719, 720, 725, 726, 728, 730–735, 740–745, 747, 753, 756–758, 760, 762, 768, 778–780, 782, 784–787, 789, 790, 799, 829, 830, 832–834, 838–840, 849, 852, 854–862, 867–869, 871–873, 875–879, 881, 884–890, 892, 893, 898, 899, 902, 903, 906–915, 919, 922, 924, 935, 936, 938, 939, 943, 948–958, 974, 980, 981, 983–985, 987–991, 996–1001, 1003, 1008, 1012, 1013, 1015–1017, 1023, 1030, 1033
\emph 211, 240, 493, 511, 745, 763, 1001, 1017
\empty 239, 337, 510, 539, 762, 793, 1017
\encodingdefault . . 26, 387, 637, 891
\end 183, 223, 279, 282–288, 290, 291, 293–318, 476, 501, 578, 599–603, 605–620, 727, 752, 831, 852, 854–874, 982, 1008
\end@doifinlist . . . 55, 403, 653, 909
\end@getprefix . . 210, 492, 744, 1000
\end@gls@islistofacronyms 21, 384, 385, 634, 888, 889
\EndAccSupp 337, 540, 793
\endcsname 15–18, 26, 41, 44, 46, 47, 52, 53, 56, 58, 59, 62, 64, 67, 72–74, 81, 82, 88–90, 92, 93, 97, 98, 100–104, 106, 107, 123–126, 137–142, 158–165, 176, 177, 182, 183, 188–191, 197, 200, 201, 203, 204, 209–211, 214–217, 219, 237, 241, 265–270, 324, 326, 330, 341, 342, 345, 349, 355–357, 374, 375, 381–383, 387, 396, 398, 399, 402, 404, 406, 408–410, 413, 414, 418, 422–425, 428–433, 443, 445–447, 454–457, 463–467, 472, 474–476, 479–481, 486, 488–490, 492, 493, 495–497, 503, 506, 507, 509, 511, 537–540, 542–545, 548, 555, 556, 573–576, 587, 592, 596, 597, 624, 625, 631–633, 637, 646, 648, 649, 652, 654, 656, 658, 659, 661, 663, 664, 668, 669, 672–676, 679–684, 694, 696–698, 706–708, 714–719, 723, 724, 726, 727, 731–733, 737, 740–742, 744–749, 755, 758–761, 763, 776, 790–793, 796–798, 801, 808–810, 826–829, 840, 845, 849, 850, 877, 878, 885–887, 891, 902–905, 908, 910, 912, 914–916, 919, 920, 924, 925, 928–932, 935–939, 941, 950, 951, 960–962, 968–973, 978, 979, 981, 982, 986–988, 993, 995–997, 999, 1000, 1002, 1003, 1005, 1015, 1017, 1031–1033
\endfoot 283–287, 289, 290, 294–296, 298, 601–604, 607–609, 854–858, 860–863
\endgroup 6, 8, 35, 203, 208, 216, 239, 376, 378, 489, 491,

510, 626, 628, 643, 741, 743, 762, 880, 882, 898, 996, 999, 1017
\endhead 283–287, 289, 290, 294–296, 298, 601–604, 607–609, 854–858, 860–863
\endtheglossary 6, 376, 626, 880
entrycounter (option) 14
\entryname 44, 47, 283, 285–287, 289, 290, 294–296, 298, 306, 308–311, 313–318, 398, 399, 601–604, 607–609, 614–616, 618–620, 648, 649, 854–858, 860–863, 867–874, 903, 905
environments:
theglossary 232
\epreto 104, 939
\equal . . 29, 43, 53, 125, 195, 389, 397, 402, 446, 484, 508, 596, 639, 647, 652, 697, 736, 760, 849, 893, 903, 908, 950, 991, 1033
esclocations (option) 12
\exp 274
\expandafter 8, 16–18, 25, 26, 41, 43, 46, 56, 58, 59, 61–64, 67, 72–75, 81, 82, 86–90, 92, 93, 97, 98, 100, 102–107, 118, 119, 123, 125–133, 166, 168, 169, 171, 176, 179, 183, 187–190, 201, 203–205, 208, 209, 212, 215, 216, 219, 224, 225, 233, 237, 239, 241, 265–270, 324, 330, 341, 342, 345, 346, 373–375, 378, 382, 383, 387, 396–399, 404, 406–410, 413–415, 418–425, 428–433, 443, 445–453, 468, 472, 473, 475, 476, 478–481, 488–493, 495, 497, 501, 502, 506, 507, 509–511, 525, 530, 537–540, 542–546, 569, 573–576, 578, 579, 592, 596, 597, 624, 627, 631–633, 636, 637, 646, 648, 649, 654, 656, 658, 659, 661, 663–665, 668, 669, 671–676, 679–684, 694, 696–704, 719–721, 724, 725, 727, 730–733, 740–745, 747, 749, 753, 754, 758, 759, 761–763, 776, 777, 783, 790–793, 796–799, 822, 826–829, 831, 832, 845, 849, 850, 877, 882, 886, 887, 891, 902–904, 910, 912–916, 919–921, 924, 925, 927–932, 935–941, 948, 950–958, 973, 974, 976, 978, 979, 982, 985–988, 995–999, 1001, 1003, 1005, 1008, 1009, 1013, 1015, 1017, 1031–1034
\expandonce . . 81, 82, 93, 127, 128, 188, 189, 204, 220, 247, 418, 419, 447, 448, 479–481, 490, 498, 506, 507, 513, 523, 525, 527, 529, 530, 533, 534, 540, 668, 669, 676, 698, 699, 731, 732, 742, 750, 758, 759, 765, 775–779, 781, 783, 786, 787, 793, 924, 925, 932, 952, 953, 986, 987, 997, 1005, 1006, 1020
\ExplSyntaxOff 6, 104, 135, 235, 236, 275, 880, 1014, 1015
\ExplSyntaxOn 4, 104, 135, 235, 236, 272, 879, 1014, 1015

F

\fi . 7, 8, 10, 11, 13–16, 18–21, 23, 25, 27, 28, 30–32, 34, 35, 37, 38, 40–43, 47–49, 52–63, 71, 74, 78, 81, 93, 97–99, 101–104, 109, 110, 120, 123–134, 169, 178–181, 184, 186, 187, 189, 190, 192–196, 201–206, 208–212, 215–217, 224, 228–231, 236–240, 251, 261, 263, 265, 279, 283, 284, 286, 289, 290, 292, 294, 296, 297, 306, 307, 309, 313, 315, 317, 320–326, 337, 346, 376–389, 391, 393–397, 399–409, 414, 416, 418, 428–432, 435, 445–457, 468, 472–474, 476, 478–482, 484, 485, 489–496, 501, 504–506, 508–511, 516, 523, 524, 526, 527, 529–537, 539, 540, 546, 575–581, 585–587, 596, 599, 601, 602, 604–608, 613–615, 617–619, 621–636, 638–640, 643–648, 650–659, 664, 666, 668, 676, 678–683, 686, 696–709, 720, 724–726, 728, 730–737, 740–748, 753, 756–758, 760–762, 768, 775, 776, 778–

780, 782, 784–790, 792, 793, 799, 828–830, 832–834, 838–840, 849, 852, 854–862, 867–869, 871–873, 875–890, 892, 893, 895–903, 905–915, 919, 920, 922, 924, 932, 934–939, 942, 943, 948–958, 974, 979–981, 983–985, 987–992, 996–1004, 1009, 1011–1013, 1015–1017, 1023, 1030, 1031, 1033
\findrootlanguage 62, 408, 658
first (key) 76
firstaccess (key) 338
\firstacronymfont 118, 119, 246, 249–251, 257, 354, 355, 364, 365, 370, 443, 513, 515, 516, 521, 525, 528, 530, 532, 555, 561, 562, 566, 693, 694, 765, 767, 768, 773, 777, 781, 783, 785, 808, 814, 815, 819, 948, 1019, 1022, 1023, 1027, 1028
firstplural (key) 76
firstpluralaccess (key) 338
\fmtversion 27, 388, 638, 892
\footnote 257, 261, 262, 370, 521, 524, 566, 773, 777, 819, 1027, 1028, 1030
footnote (acrstyle) 256, 370
footnote (option) 31
footnote-desc (acrstyle) ... 258, 372
footnote-sc (acrstyle) 258, 371
footnote-sc-desc (acrstyle) . 259, 372
footnote-sm (acrstyle) 258, 372
footnote-sm-desc (acrstyle) . 259, 372
\FootnoteNewAcronymDef ... 263, 373, 530, 531, 571, 783, 784, 825
\footnotesize 8, 378, 627, 881
\forallacronyms ... 63, 409, 659, 915
\forallglossaries .. 63, 64, 201, 214, 324, 409, 488, 494, 623, 659, 740, 746, 877, 915, 995, 1002
\forallglsentries .. 35, 64, 107, 108, 111, 178, 393, 409, 433, 434, 436, 473, 643, 659, 684–686, 724, 898, 915, 941–943, 979
\ForEachTrackedDialect 46, 328, 329, 399, 574, 591, 649, 827, 844, 905
\forglsentries 63, 64, 67, 101, 222, 324, 409, 410, 431, 500, 623, 659, 661, 682, 752, 877, 915, 916, 938, 1007
\forlistcsloop 218, 223, 497, 501, 748, 752, 1004, 1008
\forlistloop 199, 226, 487, 502, 739, 754, 994, 1010

G

\gdef ... 17, 35, 56, 72, 92, 97, 98, 202, 229, 230, 274, 382, 404, 413, 424, 428, 489, 504, 505, 596, 632, 643, 654, 663, 675, 679, 741, 756, 849, 886, 898, 910, 919, 931, 935, 996, 1012, 1033
\Genacrfullformat 118, 118, 247–251, 257, 354, 355, 364, 365, 370, 442, 443, 514–516, 521, 554, 555, 561, 562, 566, 693, 765–768, 773, 807, 808, 814, 815, 819, 947, 948, 1020, 1022, 1023, 1027
\genacrfullformat 118, 118, 247–251, 257, 354, 354, 364, 370, 442, 443, 513–516, 521, 554, 561, 562, 566, 693, 694, 765–768, 773, 807, 808, 814, 815, 819, 947, 948, 1020, 1022, 1023, 1027
\GenericAcronymFields 247, 248, 249–259, 364–370, 372, 373, 513–523, 561–563, 565, 566, 568, 765–774, 814–819, 821, 822, 1020–1027, 1029
\Genplacrfullformat 117, 119, 247, 248, 250, 251, 257, 354, 355, 364, 365, 370, 442, 443, 514–516, 521, 554, 555, 561, 562, 566, 693, 694, 766–768, 773, 807, 808, 814, 815, 819, 947, 948, 1020–1023, 1028
\genplacrfullformat ... 117, 119, 119, 247, 248, 250, 251, 257, 354, 355, 364, 365, 370, 442, 443, 514–516, 521, 554, 555, 561, 562, 566, 693, 694, 766–768, 773, 807, 808, 814, 815, 819, 947, 948, 1020–1023, 1028
\GetTitleStringDisableCommands 278, 852
\GetTitleStringSetup 278, 852

\glo@desc 581, 834
\glo@do@compare 220, 221, 498, 750, 1005, 1006
\glo@grabfirst 224, 225, 501, 502, 753, 754, 1008, 1009
\glo@label 67, 101, 410, 431, 661, 682, 916, 938
\glo@list 101, 431, 682, 938
\glo@name 233, 506, 758, 1013
\glo@parent 67, 410, 661, 916
\glo@type 101, 431, 682, 938
\glo@value 86, 420, 671, 927
\global 17, 18, 81, 84, 94, 100, 106, 107, 203, 216, 224, 225, 230, 292, 382, 383, 418, 420, 426, 430, 433, 489, 495, 501, 502, 505, 606, 632, 633, 668, 670, 677, 681, 684, 741, 747, 753, 754, 757, 859, 886, 887, 924, 926, 933, 937, 941, 996, 1003, 1009, 1012
\glolinkprefix 124, 124, 177, 232, 446, 472, 506, 697, 724, 758, 950, 979, 1013
\glosortentrieswarning 23, 23, 218, 385, 386, 496, 635, 748, 889, 890, 1004
\glossaries 272–274
\GlossariesWarning 7, 9, 23, 23, 24, 27–29, 34–36, 39, 45, 51, 54, 66, 70, 71, 78, 81, 84, 85, 111–113, 175, 190–193, 197–200, 203, 205, 210, 213, 214, 263, 337, 377, 379, 385, 386, 388, 389, 392–395, 401, 403, 410, 412, 416, 418, 436, 437, 443, 444, 471, 482, 483, 485–489, 492, 494, 495, 507, 509, 539, 575, 626–628, 635, 636, 638, 639, 642–645, 651, 653, 660, 663, 666, 668, 670, 671, 687, 694, 695, 723, 734, 735, 737–741, 744, 746, 747, 759, 761, 788, 793, 828, 880–882, 889, 890, 892, 893, 897, 898, 900, 904, 907, 909, 916, 918, 919, 922, 924, 926, 927, 944, 978, 988–990, 992–997, 1000, 1002, 1030
\GlossariesWarningNoLine 7, 23, 23, 196, 197, 201, 217, 274, 377, 385, 386, 484–486, 488, 496, 596, 626, 627, 635, 737, 738, 740, 748, 849, 880, 881, 889, 890, 992, 993, 995, 996, 1004, 1033
\glossary 38, 39, 39, 394, 395, 576, 644, 645, 829, 900
\glossaryentry . . . 209, 211, 492, 493, 576, 744, 745, 829, 999, 1000
glossaryentry (counter) 14
\glossaryentryfield 234, 363, 506, 507, 561, 581–586, 588–590, 758, 759, 814, 834–839, 841–843
\glossaryentrynumbers 11, 12, 184, 215, 216, 225, 229, 230, 379, 380, 476, 495, 502, 504, 505, 578, 629, 728, 747, 753, 754, 756, 757, 831, 883, 982, 1002, 1003, 1009, 1012
\glossaryheader . . 183, 223, 232, 275, 279, 280, 282–287, 289, 290, 293–298, 300–303, 305, 307, 308, 312, 314, 316, 319, 321–323, 325, 327, 476, 501, 506, 578, 597, 599–604, 606–615, 617–619, 621–625, 727, 752, 758, 831, 850, 852–858, 860–869, 871–879, 982, 1008, 1013
\glossarymark 51, 52, 401, 651, 652, 907
\glossaryname 19, 44, 46, 47, 383, 398, 399, 633, 648, 649, 887, 903–905
\glossarypostamble 51, 183, 223, 401, 476, 501, 578, 651, 727, 753, 831, 907, 982, 1008
\glossarypreamble 50, 183, 223, 401, 475, 500, 578, 651, 727, 752, 831, 906, 982, 1008
\glossarysection 51, 183, 223, 401, 475, 500, 578, 651, 727, 752, 831, 907, 982, 1008
\glossarystyle 237, 509, 761
glossarysubentry (counter) 15
\glossarysubentryfield 234, 363, 507, 508, 561, 581–590, 759, 814, 834–843
\glossarytitle . . . 183, 214, 215, 223, 227, 475, 495, 500, 503,

578, 727, 746, 747, 752, 755, 831, 982, 1002, 1003, 1008, 1010
\glossarytoctitle 11, 19, 20, 40, 44, 47, 51, 183, 214, 223, 227, 228, 379, 383, 384, 395, 396, 398, 399, 401, 475, 495, 500, 503, 504, 578, 629, 633, 634, 646, 648, 649, 651, 727, 746, 752, 755, 756, 831, 883, 887, 888, 901, 903, 905, 907, 982, 1002, 1008, 1011
\glossentry 101, 216, 225, 232, 275, 279–282, 284, 285, 294, 295, 297, 305, 307, 309, 312, 314, 316, 320, 321, 323, 325, 431, 495, 502, 507, 597, 599–602, 606–608, 613–615, 617–619, 621–624, 682, 747, 754, 759, 850, 852–856, 860–862, 867–869, 871–876, 878, 938, 1003, 1009, 1013
\Glossentrydesc 135, 233, 363, 507, 560, 759, 813, 1014
\glossentrydesc 135, 233, 233, 276, 279–282, 284–286, 294, 295, 297, 305, 307, 309, 312–314, 316, 320, 321, 323, 325, 326, 363, 507, 560, 598–602, 606–608, 613–615, 617–619, 621–625, 758, 813, 851–856, 860–862, 867–869, 871–873, 875–878, 1014
\Glossentryname 135, 233, 506, 758, 1014
\glossentryname 135, 232, 232, 233, 234, 276, 278–282, 284, 285, 294, 295, 297, 305, 307, 309, 312, 314, 316, 320, 321, 323, 325, 326, 362, 363, 506, 560, 597–602, 606–608, 613–615, 617–619, 621–625, 758, 813, 851–856, 860–862, 867–869, 871–873, 875, 876, 878, 1013, 1014
\glossentrynameother 135
\Glossentrysymbol 135, 233, 363, 507, 560, 759, 814, 1014
\glossentrysymbol 135, 233, 233, 276, 285, 286, 297, 309, 316, 320, 321, 323, 325, 326, 363, 507, 560, 597, 598, 602, 608, 615, 619, 621–625, 759, 814, 851, 856, 862, 869, 873, 875, 876, 878, 1014
\GLS 138, 455, 706, 960
\Gls 112, 138, 138, 241, 260, 436, 454, 511, 523, 687, 706, 763, 775, 944, 960, 1017, 1030
\gls . . . 39, 111, 137, 138, 197, 231, 241, 260, 395, 436, 454, 485, 505, 511, 523, 645, 687, 706, 737, 757, 763, 775, 900, 944, 959, 960, 992, 1013, 1017, 1030
\gls@@delimN 239, 1016
\gls@accessibility 337, 338, 540, 793, 794
\gls@accsupp@engine 337, 337, 540, 793
\gls@Alphpage 203, 204, 208, 490, 491, 741–743, 997, 999
\gls@alphpage 203, 204, 208, 490, 491, 741–743, 997, 999
\gls@arabicpage 203, 204, 208, 490, 491, 741–743, 997, 998
\gls@assign@desc 94, 94, 99, 426, 430, 677, 681, 933, 937
\gls@assign@desc@field 24
\gls@assign@descplural 524, 531, 533, 535, 569, 572, 573, 776, 783, 784, 786, 788, 823, 825, 826
\gls@assign@descplural@field . . 25
\gls@assign@field 83, 83, 88, 89, 94, 96, 98–100, 330, 419, 422, 426, 427, 429, 430, 592, 669, 673, 677–681, 845, 925, 929, 933–937
\gls@assign@firstpl 524–527, 529, 531, 533–535, 569–573, 775, 776, 778–784, 786, 788, 822–826
\gls@assign@name@field 25
\gls@assign@plural 524–527, 529, 531, 533–535, 569–573, 776, 778–784, 786, 788, 823–826
\gls@assign@symbolplural 524–527, 529, 533–535, 570, 571, 573, 776, 778–782, 786, 788, 823, 824, 826
\gls@assign@type@field 24
\gls@automake@makegloss 36, 36, 37, 38, 194, 898, 899, 991

\gls@automake@nr 36, 36, 37, 38, 193, 393, 484, 644, 735, 898, 899, 990
\gls@automake@option 36, 36, 37, 898, 899
\gls@automake@val 36, 37, 393, 644, 898, 899
\gls@begindocdefs 85, 85, 420, 671, 927
\gls@checkisacronymlist . . . 22, 123, 385, 445, 634, 696, 889, 949
\gls@checkseeallowed 77, 77, 195, 197, 416, 419, 484, 485, 666, 736, 737, 922, 991, 993
\gls@checkseeallowed@preambleonly 78, 416, 419, 666, 922
\gls@codepage 33, 62, 191, 192, 217, 392, 408, 482, 483, 496, 642, 658, 659, 734, 735, 748, 896, 914, 988, 989, 1004
\gls@debug@nr 7, 377, 395, 626, 645, 880
\gls@debug@val 7, 377, 395, 626, 645, 880
\gls@defdocnewglossaryentry 83, 85, 108, 419, 420, 434, 670, 671, 685, 926, 927, 942
\gls@defglossaryentry 83–85, 94, 95, 419, 420, 426, 670, 671, 677, 926, 927, 933
\gls@delimN 239, 239, 1016
\gls@delimR 238, 239, 1016
\gls@disablepagerefexpansion 203, 203, 208, 489–491, 741–743, 996, 997, 999
\gls@do@addxdyattribute 56, 404, 654, 910
\gls@docdef@addlabel 84, 84, 85, 670, 671, 926, 927
\gls@docdef@labellist 84, 84, 670, 926
\gls@doclearpage 53, 54, 402, 403, 652, 653, 908
\gls@doentryfmt . . . 71, 413, 663, 919
\gls@dosubst . . 127, 448, 698, 699, 952
\gls@dotoctitle 215, 227, 495, 503, 747, 755, 1002, 1003, 1010, 1011
\gls@end@sanitizesort 25, 26, 387, 636, 637, 891
\gls@endcheck 82, 419, 669, 925
\gls@glossary 39, 40, 201, 209–211, 395, 489, 492, 493, 645, 741, 744, 745, 900, 901, 996, 999, 1000
\gls@glsdefs@addlabel 84, 85, 670, 671, 926, 927
\gls@glsdefs@labellist 84, 84, 670, 926
\gls@gobbleopt 72, 73, 413, 414, 663, 664, 919, 920
\gls@grplabel . . . 271, 596, 849, 1033
\gls@hypergrouprerun 274, 274, 596, 849, 1033
\gls@if@measuring 103, 104, 939, 940
\gls@ifnotmeasuring 104, 105, 106, 203, 432, 433, 683, 684, 939–941, 996
\gls@inlinepostchild 275–277, 581, 597, 598, 834, 850, 851
\gls@inlinesep 275, 276, 581, 597, 598, 834, 850, 851
\gls@inlinesubsep 275, 276, 581, 597, 598, 834, 850, 851
\gls@islistofacronyms 21, 384, 385, 634, 888, 889
\gls@istfilebase 47, 48, 48, 191, 192, 400, 482, 483, 650, 734, 735, 905, 906, 988, 989
\gls@kernelglossredefs@nr . . 39, 900
\gls@kernelglossredefs@val . 39, 900
\gls@label 241, 511, 763, 1017
\gls@level 81, 97, 98, 225, 418, 428, 501, 502, 668, 679, 753, 924, 935, 1009
\gls@local@stepcounter 104, 104, 939
\gls@measuring@reset . . 104, 939, 940
\gls@mfirstuc@nr 41, 901, 902
\gls@mfirstuc@val 41, 901
\gls@newacronym 242, 242, 1018
\gls@noidx@nonumberlist 78, 199, 922, 995
\gls@noidxglossary 197, 200, 485, 488, 737, 740, 992, 995
\gls@nonumberlist@nr 78, 416, 666, 922
\gls@nonumberlist@val 78, 416, 666, 922
\gls@nosetquote . . . 95, 185, 187, 189, 426, 477, 478, 481, 677, 728, 730, 733, 933, 983, 985, 987
\gls@number 208, 491, 743, 998
\gls@numberedsection@nr 10, 228, 379, 503, 504, 629, 755, 883, 1011
\gls@numberedsection@val . . . 10, 228, 379, 503, 629, 755, 883, 1011

\gls@numberpage 203, 204, 208, 490, 491, 741–743, 997, 998
\gls@org@glossaryentryfield 216, 495, 747, 1003
\gls@org@glossarysubentryfield 216, 495, 747, 1003
\gls@org@insert 528, 530, 532, 780, 782, 783, 785
\gls@org@label . . . 103, 104, 939, 940
\gls@org@link 104, 939, 940
\gls@org@refstepcounter 103, 103, 104, 939, 940
\gls@org@target 104, 939, 940
\gls@orgAlph 207, 208, 491, 743, 998, 999
\gls@orgalph 207, 208, 491, 743, 998, 999
\gls@orgarabic 207, 208, 491, 743, 998
\gls@orgnumber 207, 208, 491, 743, 998
\gls@orgRoman 207, 208, 491, 743, 998, 999
\gls@orgromannumeral 207, 208, 491, 743, 998
\gls@orgthe . . . 207, 208, 491, 743, 998
\gls@original@glossary 38, 39, 394, 395, 644, 645, 900
\gls@original@makeglossary . 38, 40, 394, 395, 644, 645, 900, 901
\gls@printglossary@begin . . 215, 216
\gls@printglossary@end . . . 216, 216
\gls@protected@pagefmts 127, 203, 203, 204, 447, 490, 698, 741, 742, 952, 997
\gls@Romanpage 203, 204, 208, 490, 491, 741–743, 997, 999
\gls@romanpage 203, 204, 208, 490, 491, 741–743, 997, 998
\gls@save@numberlist 11, 12, 212, 379, 380, 494, 629, 745, 883, 1001
\gls@seenoindex@nr . . 9, 378, 628, 882
\gls@seenoindex@val . 9, 378, 628, 882
\gls@set@xr@key 77, 77, 416, 666, 922
\gls@start@measuring 104, 104, 105, 939, 940
\gls@stop@measuring . . . 104, 105, 940
\gls@suffixF 49, 49, 184, 186, 400, 476, 478, 578, 580, 650, 728–730, 832, 833, 906, 982–984
\gls@suffixFF . . . 49, 49, 184, 186, 400, 476, 478, 578–580, 650, 728, 730, 832, 833, 906, 982–984
\gls@text 118, 119, 443, 694, 948
\gls@the 208, 491, 743, 998
\gls@thissty 31, 391, 641, 895
\gls@tmp . 194, 196, 201, 263, 488, 532, 740, 785, 990, 992, 995, 1030
\gls@tmplen 134, 324, 326, 453, 587, 623–625, 705, 840, 877, 878, 959
\gls@tr@set@acronym@toctitle 20, 384, 633, 634, 888
\gls@tr@set@main@toctitle 19, 383, 633, 887
\gls@tr@set@numbers@toctitle 40, 396, 646, 901
\gls@tr@set@symbols@toctitle 40, 395, 646, 901
\gls@translate@nr . 30, 390, 640, 894
\gls@translate@val . 30, 390, 640, 894
\gls@wrglossary 202, 203, 489, 741, 996
\gls@xdy@set@lang 33, 191, 192, 217, 896, 988, 989, 1003
\gls@xdystring 127, 447, 448, 698, 699, 952
\gls@xindy@glsnumbersfalse 34, 392, 642, 897
\gls@xindy@glsnumberstrue 33, 392, 642, 896
\gls@xr@key 9, 77, 77, 78, 378, 379, 416, 628, 666, 882, 922
\glsaccessdisplay 349, 548, 801
\glsaccessibility 337, 346, 540, 545, 793, 798
\glsaccsupp 345, 346, 346, 545, 798, 799
\glsacronymtrue . . . 20, 384, 634, 888
\glsacrpluralsuffix 45, 45, 242, 252, 256–258, 398, 511, 517, 520–522, 524, 648, 763, 769, 772–774, 777, 904, 1018, 1024, 1027–1029
\glsacrshortcutsfalse 42, 397, 647, 902
\glsacrshortcutstrue 42, 396, 647, 902
\glsacspace 250, 251, 253, 515, 516, 518, 767–769, 1022–1024
\glsadd 39, 177, 178, 395, 472, 473, 645, 724, 900, 979
\glsaddall 178, 473, 724, 979
\glsaddallunused . 178, 473, 724, 979

\glsaddkey 88, 422, 673, 929
\GlsAddLetterGroup . 63, 409, 659, 915
\glsaddprotectedpagefmt 204, 205, 490, 742, 997
\GlsAddSortRule 61, 407, 408, 658, 913
\glsaddstoragekey . 87, 422, 672, 928
\GlsAddXdyAlphabet . 57, 405, 655, 911
\GlsAddXdyAttribute 55, 56, 56, 57, 404, 405, 575, 654, 655, 828, 829, 909–911
\GlsAddXdyCounters 55, 55, 56, 57, 74, 403, 404, 414, 575, 653–655, 664, 828, 909, 910, 920
\GlsAddXdyLocation 59, 406, 407, 575, 657, 829, 912, 913
\GlsAddXdyStyle . . . 61, 408, 658, 914
\glsautomakefalse 36, 37, 194, 393, 394, 484, 644, 736, 899, 990, 991
\glsautomaketrue 36, 37, 393, 644, 899
\glsautoprefix 10, 11, 228, 379, 504, 629, 756, 883, 1011
\glsbackslash 179, 473, 725, 979
\glscapitalisewords 171, 171, 721, 975, 976
\glscapscase 114–118, 137–142, 158–165, 254, 255, 350–357, 367, 368, 437, 438, 440–442, 454–457, 463–467, 519, 528, 532, 548, 549, 551–556, 564, 688, 689, 691–693, 706–708, 714–719, 771, 781, 785, 801, 802, 804–810, 817, 945–947, 959–962, 968–973, 1026
\glscategory 345, 545, 798
\glsclearpage 53, 54, 402, 403, 652, 653, 908
\glsclosebrace 60, 179, 184, 407, 473, 476, 578, 579, 657, 725, 728, 831, 832, 912, 913, 979, 982, 983
\glscompositor 48, 48, 49, 59, 186, 400, 406, 478, 578, 580, 650, 656, 729, 831, 833, 906, 912, 984
\glscounter 22, 22, 43, 55, 73, 98, 125, 385, 397, 403, 414, 429, 446, 575, 635, 647, 653, 664, 680, 697, 828, 889, 903, 909, 920, 936, 950
\glscurrententrylabel . 135, 212, 216, 494, 495, 746, 747, 1001, 1003
\glscurrentfieldvalue 69, 70, 70, 412, 413, 662, 663, 918
\glscustomtext 114, 116, 118, 137–142, 158–165, 254, 255, 350, 352, 354–357, 367, 368, 437, 440–442, 454–457, 463–467, 519, 524, 525, 528, 530, 532, 548, 551, 553–556, 564, 688, 690–693, 706–708, 714–719, 771, 777, 780–783, 785, 801, 804, 806, 808–810, 817, 818, 945–947, 959–962, 968–973, 1025, 1026
\GlsDeclareNoHyperList 22, 23, 385, 635, 889
\glsdefaultshortaccess 373, 373, 569–572, 822–825
\glsdefaulttype 19, 19, 20, 33, 50, 62, 64, 71, 72, 96, 113, 201, 213, 214, 383, 401, 408, 409, 413, 427, 437, 443, 488, 494, 495, 633, 651, 658, 659, 663, 678, 688, 694, 740, 746, 887, 896, 906, 914, 915, 919, 934, 945, 995, 1002
\glsdefmain 19, 20, 74, 383, 414, 633, 664, 887, 920
\glsdefs@newdocentry 85, 86, 671, 927
\GLSdesc 148, 459, 711, 964
\Glsdesc 148, 148, 459, 711, 964
\glsdesc 148, 148, 459, 711, 964
\GLSdescplural . . . 149, 460, 711, 965
\Glsdescplural 149, 149, 459, 711, 965
\glsdescplural 149, 149, 459, 711, 964, 965
\glsdescriptionaccessdisplay 347, 359, 363, 546, 549–551, 558, 560, 561, 800, 803, 804, 811, 813, 814
\glsdescriptionpluralaccessdisplay 348, 359, 546, 548, 549, 558, 800–802, 811
\glsdescwidth . 282, 282, 283, 284, 287, 288, 290, 291, 293, 293, 294–298, 305, 305, 306–308, 310–312, 312, 313–318, 600–603, 605–609, 613–620, 854, 855, 857–863, 866–868, 870–874
\glsdetoklabel . . . 35, 65, 65, 66–70, 78, 79, 86, 91–95, 101, 106,

107, 109, 110, 123, 166, 168, 176, 177, 197–199, 206, 208, 216, 219, 220, 224, 225, 227, 230, 231, 233, 265–269, 324, 330–332, 346–349, 374, 375, 409–412, 417, 420, 424–426, 431, 433–435, 445, 467, 468, 471, 472, 485–487, 491, 495, 497, 498, 501–503, 505, 506, 537–540, 546–548, 573, 574, 592, 624, 643, 660–662, 667, 671, 675–677, 681, 684–686, 696, 719, 723, 724, 737–739, 742, 743, 747, 749, 750, 753–755, 757, 758, 790–793, 799–801, 826, 827, 845, 846, 877, 898, 915–918, 922, 923, 927, 931–933, 938, 941–943, 950, 973, 974, 978, 979, 993, 994, 998, 999, 1003, 1005, 1006, 1009, 1010, 1012, 1013, 1031, 1032
\glsdisablehyper . 136, 454, 705, 959
\Glsdisp 142, 142, 962
\glsdisp 142, 142, 457, 708, 962
\glsdisplay 119, 437, 443, 444, 688, 694
\glsdisplayfirst 119, 437, 443, 688, 694
\glsdisplaynumberlist 175, 198, 471, 486, 723, 738, 978, 994
\glsdohyperlink 135, 136, 453, 454, 705, 706, 959
\glsdohyperlinkhook 135, 135
\glsdohypertarget 134, 136, 453, 454, 705, 706, 959
\glsdohypertargethook 134, 134
\glsdoifexists 65, 67, 69, 91–94, 105, 106, 137–143, 158–165, 171, 175–177, 198, 199, 332–336, 355–357, 362, 363, 410–412, 424–426, 432, 433, 454–457, 463–467, 469, 471, 472, 486, 487, 555, 556, 560, 593–595, 660–662, 675–677, 683, 684, 706–709, 714–718, 721, 723, 724, 738, 739, 808, 809, 813, 814, 846–849, 915–918, 931–933, 940, 941, 959–962, 968–973, 976–979, 993, 994
\glsdoifexistsordo 66, 122, 166, 410, 445, 467, 660, 695, 719, 916, 949, 973
\glsdoifexistsorwarn 66, 227, 232, 233, 410, 503, 506, 507, 660, 755, 758, 759, 916, 1010, 1013, 1014
\glsdoifnoexists 66, 83, 94, 410, 419, 426, 660, 669, 677, 916, 925, 933
\glsdonohyperlink 124, 136, 136, 446, 453, 454, 697, 705, 950, 959
\glsdoparenifnotempty 263, 532, 784, 1030
\glsdosanitizesort 16, 381, 382, 631, 885
\glsdoshowtarget 8, 134, 135, 882, 959
\glsenableentrycount 108, 434, 435, 685, 686, 942, 943
\glsenablehyper . . 136, 454, 705, 959
\glsencapwrcontent 124, 124, 950
\glsentryaccess 342, 347, 544, 546, 797, 799
\glsentrycounter 125, 236, 239, 446, 508, 510, 697, 760, 762, 950, 1015, 1017
\glsentrycounterfalse 14, 381, 631, 885
\glsentrycounterlabel 231, 231, 505, 506, 757, 758, 1013
\GlsEntryCounterLabelPrefix 228, 230, 231, 504, 505, 756, 757, 1011–1013
\glsentrycountertrue 14, 381, 631, 885
\glsentrycurrcount 109–111, 434–436, 685–687, 942–944
\Glsentrydesc 148, 169, 169, 233, 359, 363, 459, 468, 507, 558, 560, 711, 720, 759, 811, 813, 964, 974, 1014
\glsentrydesc 135, 148, 149, 169, 233, 359, 363, 438–440, 459, 468, 507, 549–551, 558, 560, 689–691, 711, 720, 758, 803, 804, 811, 813, 964, 974, 1014
\glsentrydescaccess 343, 348, 544, 546, 797, 800
\Glsentrydescplural 149, 169, 169, 359, 460, 469, 558, 711, 720, 811, 965, 975
\glsentrydescplural . . . 149, 150, 169, 359, 437, 438, 459, 460,

468, 548, 549, 558, 688, 689, 711, 720, 801, 802, 811, 965, 975
\glsentrydescpluralaccess 343, 348, 544, 547, 797, 800
\Glsentryfirst 112, 116, 144, 171, 171, 351, 358, 436, 439, 441, 458, 469, 550, 552, 557, 687, 690, 692, 709, 721, 803, 806, 810, 944, 946, 963, 975
\glsentryfirst 111, 115, 116, 144, 145, 171, 351, 352, 358, 436, 438, 439, 441, 458, 469, 550, 552, 553, 557, 687, 689, 690, 692, 709, 721, 803–806, 810, 944, 946, 963, 975
\glsentryfirstaccess 343, 347, 544, 546, 797, 799
\Glsentryfirstplural 113, 115, 146, 171, 171, 350, 358, 437, 438, 440, 458, 469, 549, 552, 557, 688, 689, 691, 710, 721, 802, 805, 810, 945, 964, 975
\glsentryfirstplural 112, 114, 115, 146, 147, 171, 350, 351, 358, 436–438, 440, 458, 459, 469, 548, 549, 551, 552, 557, 687–689, 691, 710, 721, 801, 802, 804, 805, 810, 811, 944, 945, 963, 964, 975
\glsentryfirstpluralaccess 343, 347, 544, 546, 797, 799
\glsentryfmt 73, 74, 114, 414, 437, 664, 665, 688, 920, 945
\Glsentryfull 175, 175, 248, 256, 258, 369, 371, 471, 514, 520, 521, 565, 567, 723, 766, 772, 773, 818, 820, 977, 1020, 1027, 1028
\glsentryfull . 175, 248, 256, 258, 369, 371, 471, 514, 520, 521, 565, 567, 722, 766, 772, 773, 818, 820, 977, 1020, 1027, 1028
\Glsentryfullpl 175, 175, 248, 256, 258, 369, 371, 471, 514, 520, 522, 565, 567, 723, 766, 772, 773, 819, 820, 977, 1021, 1027, 1028
\glsentryfullpl 175, 248, 256, 258, 369, 371, 471, 514, 520, 522, 565, 567, 723, 766, 772, 773, 818, 820, 977, 1021, 1027, 1028
\glsentryitem 231, 276, 278–282, 284, 285, 294, 295, 297, 305, 307, 309, 312, 314, 316, 320, 321, 323, 325, 506, 581–586, 588–590, 597, 599–602, 606–608, 613–615, 617–619, 621–624, 758, 834–843, 850, 852–856, 860–862, 867–869, 871–873, 875, 876, 878, 1013
\Glsentrylong . 112, 163, 168, 174, 174, 175, 250, 255, 256, 355, 357, 364, 368, 369, 436, 465, 468, 471, 515, 519, 520, 555, 556, 561, 564, 565, 687, 717, 719, 722, 723, 767, 771, 772, 808, 809, 814, 817, 818, 944, 971, 974, 977, 1022, 1026, 1027
\glsentrylong . 111, 118, 162, 163, 168, 174, 175, 249–259, 354, 356, 357, 364–373, 436, 443, 465, 466, 468, 471, 515–523, 530, 554, 556, 561–568, 687, 693, 717, 719, 722, 723, 767–775, 783, 808–810, 814–822, 944, 948, 971, 974, 977, 1022–1029
\glsentrylongaccess 344, 348, 544, 547, 797, 800
\Glsentrylongpl 113, 165, 174, 175, 250, 255, 256, 355, 364, 367, 369, 437, 467, 471, 515, 516, 519, 520, 555, 561, 564, 565, 688, 718, 722, 723, 767, 768, 771, 772, 808, 814, 817–819, 945, 972, 977, 1022, 1026, 1027
\glsentrylongpl 112, 119, 164, 165, 174, 175, 250, 251, 255–258, 355, 364, 365, 367, 369–371, 436, 443, 466, 467, 471, 515, 516, 519–522, 530, 536, 555, 561, 562, 564–567, 687, 694, 718, 719, 722, 723, 767, 768, 771–773, 783, 789, 808, 814, 815, 817–820, 944, 948, 972, 973, 977, 1022, 1023, 1026–1028
\glsentrylongpluralaccess 344, 348, 544, 547, 798, 800
\Glsentryname 147, 167, 167, 233, 359, 363, 459,

468, 506, 558, 560, 710, 719, 758, 811, 813, 964, 973, 974, 1014
\glsentryname 135, 147, 148, 167, 212, 324, 359, 362, 459, 468, 557, 558, 560, 624, 710, 711, 719, 811, 813, 877, 964, 973, 974, 1001
\glsentrynumberlist 175, 175, 176, 198, 471, 486, 723, 738, 977, 978, 993
\glsentryparent . . 172, 470, 721, 976
\Glsentryplural 115, 145, 170, 170, 350, 358, 438, 440, 458, 469, 548, 551, 557, 688, 691, 710, 720, 802, 805, 810, 945, 963, 975
\glsentryplural 114, 115, 145, 146, 170, 350, 351, 358, 437, 438, 440, 458, 469, 548, 549, 551, 552, 557, 688, 689, 691, 710, 720, 801, 802, 804, 805, 810, 945, 963, 975
\glsentrypluralaccess 343, 347, 544, 546, 797, 799
\Glsentryprefix 331, 331, 334, 592, 594, 845, 847
\glsentryprefix . . 330, 331, 332, 335, 592, 593, 595, 845, 846, 848
\Glsentryprefixfirst 330, 331, 334, 592, 594, 845, 847
\glsentryprefixfirst 330, 331, 332, 335, 592, 593, 595, 845, 846, 848
\Glsentryprefixfirstplural 331, 331, 335, 592, 595, 845, 848
\glsentryprefixfirstplural 330, 331, 333, 336, 592, 593, 596, 845, 847, 849
\Glsentryprefixplural 331, 331, 334, 592, 594, 845, 848
\glsentryprefixplural 330, 331, 333, 336, 592, 593, 595, 845, 847, 849
\glsentryprevcount 109, 110, 434, 435, 685, 686, 942, 943
\Glsentryshort 117, 159, 168, 174, 174, 251, 257, 258, 353, 355, 365, 370, 371, 442, 463, 468, 470, 516, 521, 554, 555, 562, 566, 567, 693, 715, 720, 722, 768, 773, 807, 808, 815, 819, 820, 947, 969, 974, 977, 1023, 1027, 1028
\glsentryshort . . . 117, 118, 158, 159, 168, 174, 175, 248–259, 353–356, 364–373, 442, 443, 463, 464, 468, 470, 471, 514–523, 553–556, 561–568, 693, 714, 715, 720, 722, 723, 766–775, 807–809, 814–822, 947, 948, 968, 969, 974, 977, 1021–1029
\glsentryshortaccess 343, 348, 544, 547, 797, 800
\Glsentryshortpl 116, 161, 174, 174, 251, 257, 258, 353, 365, 370, 371, 441, 464, 471, 516, 521, 522, 553, 562, 566, 567, 692, 716, 722, 768, 773, 806, 815, 819, 820, 946, 970, 977, 1023, 1028
\glsentryshortpl 116, 117, 119, 160, 161, 174, 175, 250, 251, 256–258, 352, 353, 355, 364, 365, 369–371, 441, 443, 464, 465, 471, 515, 516, 520–522, 536, 553, 555, 561, 562, 565–567, 692, 694, 716, 722, 723, 767, 768, 772, 773, 789, 806, 808, 814, 815, 818–820, 946, 948, 970, 977, 1022, 1023, 1027, 1028
\glsentryshortpluralaccess 343, 348, 544, 547, 797, 800
\glsentrysort 172, 470, 721, 976
\Glsentrysymbol 150, 170, 170, 233, 360, 363, 460, 469, 507, 558, 561, 712, 720, 759, 811, 814, 965, 975, 1014
\glsentrysymbol 135, 150, 151, 170, 233, 360, 363, 438–440, 460, 469, 507, 525, 528, 532, 549–551, 558, 560, 689–691, 711, 712, 720, 759, 777, 781, 785, 803, 804, 811, 814, 965, 975, 1014
\glsentrysymbolaccess 343, 347, 544, 546, 797, 799
\Glsentrysymbolplural 151, 170, 170, 360, 460, 469, 558, 712, 721, 812, 966, 975
\glsentrysymbolplural . 151, 152, 170, 360, 437, 438, 460, 469, 525, 528, 532, 548, 549, 558,

688, 689, 712, 721, 777, 781, 785, 801, 802, 812, 965, 966, 975
\glsentrysymbolpluralaccess 343, 347, 544, 546, 797, 800
\Glsentrytext . 115, 144, 170, 170, 351, 357, 439, 441, 457, 469, 550, 552, 557, 690, 692, 709, 720, 803, 805, 810, 946, 962, 975
\glsentrytext 115, 116, 143, 169, 177, 212, 351, 352, 357, 438, 439, 441, 457, 469, 472, 494, 549, 550, 552, 553, 557, 689–692, 709, 720, 724, 745, 803, 805, 806, 810, 946, 962, 975, 978, 1001
\glsentrytextaccess 342, 347, 544, 546, 797, 799
\glsentrytitlecase 171, 469, 470, 721, 976
\glsentrytype 96, 172, 427, 470, 678, 721, 934, 976
\Glsentryuseri 152, 172, 172, 360, 461, 470, 559, 712, 722, 812, 966, 976
\glsentryuseri 152, 153, 172, 360, 460, 461, 470, 559, 712, 721, 812, 966, 976
\glsentryuseriaccess 344, 348, 544, 547, 798, 800
\Glsentryuserii 153, 172, 172, 361, 461, 470, 559, 713, 722, 812, 966, 976
\glsentryuserii 153, 154, 172, 361, 461, 470, 559, 712, 713, 722, 812, 966, 976
\glsentryuseriiaccess 344, 349, 544, 547, 798, 800
\Glsentryuseriii 154, 173, 173, 361, 461, 470, 559, 713, 722, 812, 967, 976
\glsentryuseriii 154, 155, 173, 361, 461, 470, 559, 713, 722, 812, 967, 976
\glsentryuseriiiaccess 344, 349, 545, 547, 798, 800
\Glsentryuseriv 155, 173, 173, 361, 462, 470, 559, 713, 722, 812, 967, 976
\glsentryuseriv 155, 156, 173, 361, 462, 470, 559, 713, 722, 812, 813, 967, 976
\glsentryuserivaccess 344, 349, 545, 547, 798, 801
\Glsentryuserv 156, 173, 173, 362, 462, 470, 559, 714, 722, 813, 968, 977
\glsentryuserv 156, 157, 173, 362, 462, 470, 559, 560, 714, 722, 813, 967, 968, 976, 977
\glsentryuservaccess 344, 349, 545, 547, 798, 801
\Glsentryuservi 157, 174, 174, 362, 462, 470, 560, 714, 722, 813, 968, 977
\glsentryuservi 157, 158, 173, 362, 462, 463, 470, 560, 714, 722, 813, 968, 977
\glsentryuserviaccess 344, 349, 545, 548, 798, 801
\glsesclocationsfalse 12, 197, 485, 737, 884, 992
\glsesclocationstrue 380, 629
\glsexpandfields . . 83, 419, 669, 925
\glsfieldaccsupp 345, 345, 346, 545, 546, 798, 799
\glsfielddef 92, 425, 675, 931
\glsfieldedef 91, 424, 675, 931
\glsfieldfetch 92, 171, 425, 469, 676, 721, 932, 976
\glsfieldgdef 91, 424, 675, 931
\glsfieldxdef 91, 424, 675, 931
\glsfindwidesttoplevelname 324, 623, 877
\GLSfirst 144, 458, 709, 963
\Glsfirst 144, 144, 458, 709, 963
\glsfirst 144, 144, 458, 709, 962, 963
\glsfirstaccessdisplay 347, 351, 352, 358, 546, 550, 552, 553, 557, 799, 803–806, 810
\GLSfirstplural . . 146, 458, 710, 964
\Glsfirstplural 146, 146, 458, 710, 963
\glsfirstplural 146, 146, 458, 710, 963
\glsfirstpluralaccessdisplay 347, 350, 358, 546, 548, 549, 551, 552, 557, 799, 801, 802, 804, 805, 810, 811
\glsfirstpluralacessdisplay 351, 552, 805
\glsgenacfmt 116, 249–251, 257, 352, 364, 370, 441, 515,

516, 521, 553, 561, 562, 566, 692, 767, 768, 772, 806, 814, 815, 819, 946, 1021–1023, 1027
\glsgenentryfmt 114, 114, 249–251, 255, 257, 261, 350, 364, 368, 370, 440, 515, 516, 519, 521, 523, 525, 526, 528, 530, 532, 534, 536, 551, 561, 562, 564, 566, 691, 767, 768, 771, 772, 775, 777, 779, 780, 783, 785, 787, 789, 804, 814, 815, 818, 819, 945, 1021–1023, 1026, 1027, 1030
\glsgetgrouplabel 236, 508, 760, 1015
\glsgetgrouptitle 235, 275, 278–280, 299–304, 320–324, 327, 508, 597, 599, 600, 610–613, 621–623, 625, 760, 850, 852, 853, 863–866, 875–877, 879, 1014, 1034
\glsglossarymark 51, 51, 401, 651, 907
\glsgroupheading 184, 225, 235, 275, 279, 280, 282, 284, 285, 293, 295, 297, 299–305, 307, 309, 312, 314, 316, 319–325, 327, 476, 502, 508, 579, 597, 599–602, 606–608, 610–615, 617–619, 621–625, 728, 754, 760, 832, 850, 852–856, 860–869, 871–879, 983, 1009, 1014
\glsgroupskip . 183, 184, 225, 234, 276, 279, 283, 284, 286, 289, 290, 294–297, 306, 307, 309, 313, 315, 317, 320, 322, 323, 326, 476, 502, 508, 578, 579, 598, 599, 601, 602, 604, 607, 608, 613–615, 617–619, 621–623, 625, 728, 754, 760, 831, 832, 851, 852, 854–858, 860–862, 867–869, 871–873, 875–877, 879, 982, 983, 1009, 1014
\glshyperfirstfalse 257, 370, 521, 566, 773, 819, 1027
\glshyperfirsttrue . 31, 391, 641, 895
\glshyperlink 176, 212, 472, 494, 724, 745, 978, 1001
\glshypernavsep 274, 274, 275, 597, 850, 1034
\glshypernumber 50, 238, 240, 400, 509–511, 651, 761–763, 906, 1016, 1017
\glsifhyper 121, 444, 695
\glsifhyperon 123, 444, 445, 695, 696, 950
\glsIfListOfAcronyms 21, 21, 22, 384, 385, 634, 888, 889
\glsifmeasuring .. 104, 104, 939, 940
\glsifplural 114, 116, 117, 137–142, 158–165, 254, 350, 352, 353, 355–357, 367, 437, 440–442, 454–457, 463–467, 519, 525, 528, 530, 532, 548, 551, 553–556, 564, 688, 691–693, 706–708, 714–719, 771, 777, 780, 783, 785, 801, 804, 806–810, 817, 945–947, 959–962, 968–973, 1026
\glsifusedtranslatordict 29, 390, 640, 894
\glsifusetranslator 29–31, 46, 47, 328, 390, 391, 398, 399, 574, 591, 639–641, 648, 649, 827, 844, 894, 895, 904, 905
\glsignore 178, 473, 724, 979
\glsindexingsetting 15, 18, 194, 196, 885, 887, 991, 992
\glsindexonlyfirstfalse 31, 391, 641, 895
\glsinlinedescformat 276, 277, 581, 598, 834, 851
\glsinlinedopostchild 275, 276, 581, 597, 598, 834, 850, 851
\glsinlineemptydescformat 276, 277, 581, 597, 598, 834, 851
\glsinlineifhaschildren .. 276, 276
\glsinlinenameformat 276, 277, 581, 597, 598, 834, 851
\glsinlineparentchildseparator 276, 277, 581, 598, 834, 851
\glsinlinepostchild 276, 277, 581, 598, 834, 851
\glsinlineseparator 276, 277, 581, 598, 834, 851
\glsinlinesubdescformat 276, 277, 581, 598, 834, 851, 852
\glsinlinesubnameformat 276, 277, 581, 598, 834, 851, 852
\glsinlinesubseparator 276, 277, 581, 598, 834, 851

\glsinsert . . . 114–118, 137–142, 158–165, 255, 350–357, 367, 368, 437–442, 454–457, 463–467, 519, 525, 528, 530, 532, 548–556, 564, 688–693, 706–708, 714–719, 771, 777, 780–783, 785, 801–810, 817, 818, 945–947, 960–962, 968–973, 1026
\glskeylisttok 246, 247, 373, 513, 524–527, 529, 531, 533–537, 569–573, 765, 775, 776, 778–784, 786–788, 790, 822–826, 1019, 1020
\glslabel 95, 114–118, 122–124, 159–165, 249–251, 254, 255, 257, 350–354, 364, 367, 368, 370, 427, 437–442, 445, 446, 463–467, 515, 516, 519, 521, 525, 528, 530, 532, 548–554, 561, 562, 564, 566, 677, 688–693, 696, 697, 715–719, 767, 768, 771, 772, 777, 780, 781, 783, 785, 801–807, 814, 815, 817, 819, 933, 945–947, 949, 950, 969–973, 1021–1023, 1026, 1027
\glslabelhypertarget 134
\glslabelhypertargetdefs . . 135, 135
\glslabelhypertargetprefix 135, 135
\glslabelhypertargetvalue . 135, 135
\glslabeltok 246, 247, 513, 523–537, 569–572, 765, 775–785, 787–790, 822–825, 1019, 1020
\glsletentryfield 166, 467, 719, 973
\Glslink 122, 122, 949
\glslink 122, 122, 247, 248, 255–257, 261, 368–371, 444, 513, 514, 520, 521, 524, 565–567, 695, 765, 766, 771–773, 777, 818, 820, 949, 1020, 1026–1028, 1030
\glslinkcheckfirsthyperhook 123, 123, 445, 696, 949, 950
\glslinkpostsetkeys 123, 124, 445, 446, 696, 950
\glslinkvar 121, 121, 444, 695, 948, 949
\glslistdottedwidth 281, 281, 582, 600, 835, 853
\glslistexpandedname . . 278, 278, 852
\glslistgroupheaderfmt . . . 278, 278, 279, 280, 599, 600, 852, 853
\glslistinit 278, 279, 852
\glslistnavigationitem 278, 279, 280, 599, 852, 853
\glslocalreset 105, 107, 432, 433, 683, 684, 940, 941
\glslocalresetall 107, 433, 684, 941
\glslocalunset 106, 108, 120, 433, 434, 454–457, 684, 685, 706–708, 941, 942, 948
\glslocalunsetall 108, 434, 685, 942
\glslocationcstoencap 50, 906
\glslongaccessdisplay . 348, 354–357, 364–373, 547, 554–556, 561–569, 800, 808–810, 814–822
\glslongaccesskey 373, 573, 826
\glslongkey 242, 373, 511, 573, 763, 826, 1018
\glslongpluralaccessdisplay 348, 355, 364, 365, 367, 369–371, 547, 555, 561, 562, 564–567, 569, 800, 808, 814, 815, 817–820, 822
\glslongpluralaccesskey 374, 573, 826
\glslongpluralkey 243, 374, 511, 573, 763, 826, 1018
\glslongtok 246, 247–251, 255, 257, 364, 368, 370, 373, 513–516, 520, 521, 523–531, 533–537, 561, 562, 565, 566, 569–572, 765–768, 771, 773, 775–784, 786–790, 814, 815, 818, 819, 822–826, 1020–1023, 1026, 1027
\glslowercase 4, 879
\glsLTpenaltycheck 292, 292, 605, 606, 859
\glsmakefirstuc 41, 902
\glsmcols 299, 299, 300–304, 609–612, 863–866
\glsmeasuredepth 105, 940
\glsmeasureheight 105, 134, 940, 959
\glsmeasurewidth . 105, 324–326, 940
\glsmfuaddmap 5, 90, 111, 113, 122, 138, 140, 142, 144–158, 160, 162, 164, 233, 244, 245, 331, 333, 334, 879, 880, 930, 931, 944, 949, 960–968, 970–972, 974–977, 1014, 1019
\glsmfublocker 5, 5, 90, 138, 141, 143–157, 159, 161, 163, 165, 879, 880, 931, 960–972

\glsmfuexcl 5, 5, 167, 169–175, 178, 879, 880, 979
\glsmoveentry 101, 431, 681, 938
\GLSname 147, 459, 710, 964
\Glsname 147, 147, 459, 710, 964
\glsname 147, 147, 459, 710, 964
\glsnameaccessdisplay 346, 359, 362, 363, 546, 557, 558, 560, 561, 799, 811, 813, 814
\glsnamefont 233, 238, 362, 363, 506, 507, 509, 540, 560, 758, 759, 761, 793, 813, 1013, 1014, 1016
\glsnavhypergroupdotarget . 273, 273
\glsnavhyperlink 271, 274, 275, 596, 597, 849, 850, 1033, 1034
\glsnavhyperlinkname 271, 272, 273, 596, 849, 1033
\glsnavhypertarget 273, 278–280, 300–304, 321, 322, 324, 327, 596, 599, 600, 610–613, 622, 623, 625, 849, 852, 853, 863–866, 875–877, 879, 1033
\glsnavigation 274, 278–280, 300–304, 321–323, 327, 596, 599, 610–612, 622, 623, 625, 850, 852, 853, 863–866, 875–877, 879, 1033
\glsnavigationitem 274, 274
\glsnextpages 12, 78, 215, 230, 380, 416, 495, 505, 629, 666, 747, 757, 884, 922, 1003, 1012
\glsnoexpandfields . 83, 419, 669, 925
\glsnogroupskipfalse 14, 380, 630, 884
\glsnoidxdisplayloc 199, 200, 226, 487, 488, 503, 739, 740, 755, 994, 995, 1010
\glsnoidxdisplayloclisthandler 199, 226, 487, 502, 739, 754, 994, 1010
\glsnoidxloclist 198, 225, 226, 486, 502, 738, 753, 754, 993, 1009, 1010
\glsnoidxloclisthandler 226, 226, 502, 754, 1010
\glsnoidxnumberlistloophandler 199, 200, 487, 488, 739, 994, 995
\glsnoidxprenumberlist 78, 225, 922, 1009
\glsnoidxstripaccents 26, 26, 387, 388, 637, 638, 891, 892
\glsnomakeindexwarning . 54, 186, 187, 403, 478, 653, 730, 909, 985
\glsnonextpages 78, 215, 230, 416, 495, 505, 666, 747, 757, 922, 1003, 1012
\glsnopostdotfalse . 13, 380, 630, 884
\glsnoxindywarning . . . 49, 54, 55, 57, 58, 60–63, 180, 181, 400, 403–405, 407, 408, 473, 474, 650, 653–655, 657–659, 725, 726, 906, 909–911, 913, 914, 980
\glsnumberformat 49, 176, 400, 471, 472, 651, 723, 906, 978
\glsnumberlistloop 199, 200, 487, 739, 994, 995
\glsnumbersgroupname . 40, 44, 47, 236, 396, 398, 399, 508, 646, 648, 650, 760, 901, 904, 905, 1015
\glsnumlistlastsep 176, 176, 198, 472, 487, 724, 738, 978, 994
\glsnumlistparser 176, 195, 472, 485, 724, 736, 978, 992
\glsnumlistsep 176, 176, 198, 472, 487, 724, 738, 978, 994
\glsopenbrace 60, 179, 184, 407, 473, 476, 578, 579, 657, 724, 728, 831, 832, 912, 913, 979, 982, 983
\glsorder 32, 32, 191–193, 195, 221, 392, 482–484, 499, 641, 642, 733–736, 751, 896, 988–991, 1006
\glsorg@endtheglossary 6, 6, 376, 626, 880
\glsorg@PrintChanges 6, 376, 626, 880
\glsorg@theglossary 6, 6, 376, 626, 880
\glspagelistwidth 282, 284, 287, 288, 290, 291, 293, 295–298, 305, 307, 308, 310, 311, 312, 314–318, 600–603, 605–609, 613–620, 854, 855, 857–863, 867, 868, 870–874
\glspar 47, 400, 650, 905
\glspatchLToutput 288–291, 292, 604, 605, 857–859
\glspatchtabularx 105, 432, 683, 940
\glspenaltygroupskip 289, 290, 292, 604, 605, 857–859
\glspercentchar 86, 179, 183, 185, 420, 421, 473, 476, 477, 671, 672, 725, 727, 729, 927, 928, 979, 982, 983

\GLSpl 141, 456, 708, 961
\Glspl 113, 140, 140, 261, 437, 456, 523, 687, 707, 775, 945, 961, 1030
\glspl 112, 139, 140, 260, 436, 455, 523, 687, 707, 775, 944, 960, 961, 1030
\GLSplural 145, 458, 710, 963
\Glsplural . . . 145, 145, 458, 710, 963
\glsplural . . . 145, 145, 458, 710, 963
\glspluralaccessdisplay 347, 350, 351, 358, 546, 548, 549, 551, 552, 557, 799, 801, 802, 804, 805, 810
\glspluralsuffix 45, 45, 98, 247, 250, 251, 364, 365, 369–372, 398, 429, 515, 516, 561, 562, 566, 568, 648, 679, 680, 767, 768, 815, 819–821, 904, 936, 1020, 1022, 1023
\glspostdescription . 13, 47, 277, 279, 280, 282, 294, 305, 312, 313, 320, 321, 323, 325, 326, 380, 400, 581–589, 598–601, 606, 613, 617, 621–625, 630, 650, 835, 837–843, 851–854, 860, 867, 871, 875–878, 884, 905
\glspostinline 275, 277, 597, 598, 850, 851
\glspostlinkhook 122, 122, 138–143, 158–165, 355–357, 445, 454–457, 463–467, 555–557, 696, 706–709, 715–719, 808–810, 949, 960–962, 968–973
\glsprefixsep 332, 332, 333–336, 593–596, 846–849
\glsprestandardsort 15, 16, 381, 382, 631, 885, 886
\glsquote 179, 473, 725, 979
\glsrefentry 231, 505, 757, 1013
\glsreset 105, 107, 432, 433, 683, 684, 940, 941
\glsresetall 107, 433, 684, 941
\glsresetcurrcountfalse . . 108, 942
\glsresetentrycounter 230, 505, 757, 1012
\glsresetentrylist 183, 223, 229, 230, 230, 476, 501, 504, 505, 578, 728, 752, 756, 757, 831, 982, 1008, 1012
\glsresetsubentrycounter 230, 231, 276, 505, 506, 581, 598, 757, 758, 834, 851, 1012, 1013
\glsrestoreLToutput . . . 292, 605, 859
\glssanitizesortfalse 28, 389, 639, 893
\glssanitizesorttrue 28, 389, 638, 639, 893
\glssavenumberlistfalse 12, 380, 629, 883
\glssavewritesfalse 38, 394, 644, 900
\glssee 211, 493, 745, 1001
\glsseeformat 182, 197, 199, 211, 475, 486, 487, 493, 578, 727, 737, 739, 745, 831, 981, 993, 994, 1001
\glsseeitem 212, 212, 493, 494, 745, 1001
\glsseeitemformat 212, 212, 494, 745, 1001
\glsseelastsep 212, 212, 493, 745, 1001
\glsseelist . . 211, 211, 493, 745, 1001
\glsseesep . . 212, 212, 493, 745, 1001
\glssentencecase 4, 118, 119, 122, 142, 166, 168, 169, 879, 948, 949, 962, 973, 974
\glsSetAlphaCompositor 49, 400, 650, 906
\glsSetCompositor . 48, 400, 650, 906
\glssetexpandfield 24, 24, 27–29, 386, 388, 389, 636, 638, 639, 890, 892, 893
\glssetnoexpandfield 24, 24, 25, 27, 28, 386, 388, 389, 636, 638, 639, 890, 892, 893
\GlsSetQuote 95, 185, 186, 426, 477, 478, 481, 677, 728, 730, 733, 933, 983, 985, 987
\glsSetSuffixF 49, 400, 650, 906
\glsSetSuffixFF . . . 49, 400, 650, 906
\glssettoctitle 44, 47, 215, 398, 399, 495, 648, 649, 747, 903, 905, 1002
\glssetwidest 324, 624, 877
\GlsSetWriteIstHook 186, 478, 730, 985
\GlsSetXdyCodePage 62, 408, 658, 659, 914
\GlsSetXdyFirstLetterAfterDigits 179, 473, 725, 979, 980
\GlsSetXdyLanguage . 62, 408, 658, 914

\GlsSetXdyLocationClassOrder 61, 407, 657, 913
\GlsSetXdyMinRangeLength 180, 474, 725, 726, 980
\GlsSetXdyNumberGroupOrder 180, 473, 474, 725, 980
\GlsSetXdyStyles . . 61, 408, 658, 914
\glsshortaccessdisplay 168, 348, 353–356, 364–373, 468, 547, 553–556, 561–569, 720, 800, 807–809, 814–822, 974
\glsshortaccesskey 373, 573, 826
\glsshortaccsupp . 346, 346, 545, 798
\glsshortkey 242, 373, 511, 573, 763, 826, 1018
\glsshortplaccsupp 346, 545, 798
\glsshortpluralaccessdisplay 348, 352, 353, 355, 364, 365, 369–371, 547, 553, 555, 561, 562, 565–567, 569, 800, 806, 808, 814, 815, 818–820, 822
\glsshortpluralaccesskey 373, 573, 826
\glsshortpluralkey 242, 373, 511, 573, 763, 826, 1018
\glsshorttok . . 246, 247, 373, 513, 523–531, 533–537, 569–573, 765, 775–790, 822–826, 1019, 1020
\glsshowaccsupp 7, 9, 377, 378, 627, 628, 881, 882
\glsshowtarget 8, 8, 377, 378, 627, 881, 882
\glsshowtargetfont 8, 8, 377, 378, 627, 881
\glsshowtargetfonttext . . . 8, 8, 881
\glsshowtargetinner 8, 8, 881
\glsshowtargetouter 8, 8, 378, 627, 881
\glsshowtargetsymbol 8, 8, 378, 627, 881
\glssortnumberfmt 16, 17, 18, 382, 383, 631, 632, 886, 887
\glsspace 243, 243, 512, 763, 776, 1018
\glsstepentry 230, 231, 505, 506, 757, 758, 1012, 1013
\glsstepsubentry 230, 231, 505, 506, 757, 758, 1012, 1013
\glssubentrycounterfalse 15, 381, 631, 885
\glssubentrycounterlabel 231, 231, 505, 506, 757, 758, 1013
\glssubentryitem 231, 276, 279–282, 284, 286, 294, 295, 297, 305, 307, 309, 313, 314, 316, 320, 321, 323, 325, 506, 581–590, 598–602, 606–608, 613–615, 617–619, 621–624, 758, 834–843, 851–856, 860–862, 867–869, 871–873, 875, 876, 878, 1013
\GLSsymbol 150, 460, 712, 965
\Glssymbol . . . 150, 150, 460, 711, 965
\glssymbol . . . 150, 150, 460, 711, 965
\glssymbolaccessdisplay . . 347, 360, 363, 546, 549–551, 558, 560, 561, 799, 803, 804, 811, 814
\glssymbolnav . . . 275, 597, 850, 1034
\GLSsymbolplural . 151, 460, 712, 966
\Glssymbolplural 151, 151, 460, 712, 965
\glssymbolplural 151, 151, 460, 712, 965
\glssymbolpluralaccessdisplay 347, 360, 546, 548, 549, 558, 799, 801, 802, 812
\glssymbolsgroupname . 40, 44, 47, 236, 395, 398, 399, 508, 646, 648, 650, 760, 901, 904, 905, 1015
\glstarget 104, 232, 277–282, 284–286, 294, 295, 297, 305, 307, 309, 312–314, 316, 320, 321, 323, 325, 326, 506–508, 581–590, 598–602, 606–608, 613–615, 617–619, 621–625, 758–760, 834–843, 851–856, 860–862, 867–869, 871–873, 875–878, 939, 940, 1013
\glstexorpdfstring 167, 167, 171, 205, 973, 976, 998
\GLStext 143, 457, 709, 962
\Glstext 143, 144, 457, 709, 962
\glstext 143, 144, 457, 709, 962
\glstextaccessdisplay . 347, 351, 352, 357, 546, 549, 550, 552, 553, 557, 799, 803, 805, 806, 810
\glstextformat 114, 122, 124, 437, 445, 446, 688, 696, 697, 945, 949, 950
\glstextup 45, 242, 371, 398, 511, 568, 648, 763, 821, 904, 1018
\glstildechar 56, 179, 183, 184, 404, 473, 476, 654,

725, 727, 728, 910, 979, 982, 983
\glstranslatefalse 30, 31, 390, 391, 640, 894, 895
\glstranslatetrue 30, 31, 390, 391, 640, 641, 894, 895
\glstreechildpredesc 319, 320, 321, 621, 622, 874–876
\glstreegroupheaderfmt 299–304, 318, 320–324, 327, 610–613, 620–623, 625, 863–866, 874–877, 879
\glstreeindent 321, 322, 323, 325, 326, 585–587, 622–625, 839, 840, 876, 878
\glstreeitem . . 299, 300, 319, 319, 609, 610, 620, 621, 863, 864, 874
\glstreenamebox 324, 325, 326, 624, 625, 877, 878
\glstreenamefmt . . 318, 318, 320, 321, 323–326, 620–625, 874–878
\glstreenavigationfmt 300–304, 318, 321–323, 327, 610–612, 620, 622, 623, 625, 863–866, 874–877, 879
\glstreepredesc 319, 320, 321, 323, 621–623, 874–876
\glstreesubitem 299, 319, 319, 609, 621, 863, 874
\glstreesubsubitem 299, 319, 319, 609, 621, 863, 874
\glstype 123, 124, 137–142, 158–165, 355–357, 445, 454–457, 463–467, 555, 556, 696, 706–708, 714–719, 808–810, 949, 950, 960–962, 968–973
\glsucmarkfalse . . . 14, 380, 630, 884
\glsucmarktrue 14, 380, 630, 884
\glsunexpandedfieldvalue 92, 278, 676, 852, 932
\glsunset 106, 107, 109, 110, 120, 432–435, 454–457, 683–686, 706–709, 940, 941, 943, 948
\glsunsetall 107, 434, 684, 941
\glsupacrpluralsuffix 45, 252, 258, 398, 517, 522, 526, 529, 531, 534, 648, 769, 774, 779, 782, 784, 787, 904, 1023, 1024, 1029
\glsuppercase . 4, 52, 90, 115–118, 143, 145–159, 161, 163, 165, 247, 248, 255–257, 335, 336, 351–354, 356–362, 367–369, 371, 879, 907, 930, 945–947, 962–971, 973, 1020, 1026–1028
\GlsUseAcrEntryDispStyle 248, 249, 251–254, 256, 258, 259, 365–367, 369, 371, 372, 514–520, 522, 562, 563, 566–568, 766–770, 772, 774, 815–817, 819, 821, 1021, 1023–1025, 1027–1029
\GlsUseAcrStyleDefs 248, 249, 251–254, 256, 258, 259, 365–367, 369, 371–373, 514, 515, 517–520, 522, 523, 562, 563, 566–568, 766–772, 774, 815–817, 819, 821, 822, 1021, 1023–1025, 1027–1029
\GLSuseri 152, 461, 712, 966
\Glsuseri 152, 152, 461, 712, 966
\glsuseri 152, 152, 460, 712, 966
\glsuseriaccessdisplay 348, 360, 547, 559, 800, 812
\GLSuserii 153, 461, 713, 966
\Glsuserii . . . 153, 153, 461, 713, 966
\glsuserii . . . 153, 153, 461, 712, 966
\glsuseriiaccessdisplay 348, 361, 547, 559, 800, 812
\GLSuseriii 154, 461, 713, 967
\Glsuseriii . . . 154, 154, 461, 713, 967
\glsuseriii 154, 154, 461, 713, 966, 967
\glsuseriiiaccessdisplay 349, 361, 547, 559, 800, 812
\GLSuseriv 155, 462, 713, 967
\Glsuseriv . . . 155, 155, 462, 713, 967
\glsuseriv . . . 155, 155, 461, 713, 967
\glsuserivaccessdisplay 349, 361, 547, 559, 800, 812, 813
\GLSuserv 156, 462, 714, 968
\Glsuserv 156, 156, 462, 714, 967
\glsuserv 156, 156, 462, 713, 967
\glsuservaccessdisplay 349, 362, 547, 559, 560, 801, 813
\GLSuservi 157, 462, 714, 968
\Glsuservi . . . 157, 157, 462, 714, 968
\glsuservi . . . 157, 157, 462, 714, 968
\glsuserviaccessdisplay 349, 362, 547, 560, 801, 813
\GlsWarnAddProtectedPageFmt 204, 205, 997

\glswrallowprimitivemodsfalse 206, 998
\glswrallowprimitivemodstrue 491, 743
\glswrglossdisableanchorcmds 205, 209, 239, 998, 999, 1017
\glswrglossdisablelocationcmds 200, 203, 204, 995, 997
\glswrglosslocationtarget 205, 239, 998, 1017
\glswrglosslocationtextfmt 205, 240, 998, 1017
\glswrite 181–186, 194, 196, 201, 474–478, 484, 488, 489, 576–580, 726–730, 736, 740, 741, 830–833, 980–984, 991, 996
\glswritedefhook 86, 94, 421, 426, 671, 677, 927, 933
\glswriteentry 203, 203, 489, 741, 996, 997
\glswritefiles 38, 201, 394, 488, 644, 740, 899, 900, 995
\glsxindyfalse 33, 392, 642, 896
\glsxindytrue 33, 34, 392, 642, 896, 897

H

\H 26, 388, 637, 892
\hangindent 303, 304, 319, 321, 323, 325–327, 585–587, 612, 613, 620, 622–625, 838–840, 866, 874–876, 878, 879
\hbox 105, 281, 432, 582, 600, 683, 835, 853, 940
\hfill 281, 582, 600, 835, 853
\hline 283–287, 294–296, 298, 306–308, 310, 311, 313–318, 601–603, 607–609, 613–620, 854–856, 860–863, 867–874
\hsize 281, 282, 293, 305, 312, 600, 606, 613, 617, 853, 854, 860, 866, 867, 870, 871
\hspace 319, 621, 874
\hss 281, 582, 600, 835, 853
\ht 292, 605, 859
\hyperbf 240, 511, 763, 1017
\hyperdef 43, 397, 647, 903
\hyperemph 240, 511, 763, 1017
hyperfirst (option) 31
\hyperit 240, 511, 763, 1017
\hyperlink 120, 135, 240, 444, 453, 510, 695, 705, 762, 948, 959, 1017
\hypermd 240, 511, 763, 1017
\hyperrm 240, 511, 762, 1017
\hypersc 240, 511, 763, 1017
\hypersf 240, 511, 763, 1017
\hypersl 240, 511, 763, 1017
\hypertarget 134, 453, 705, 959
\hypertt 240, 511, 763, 1017
\hyperup 240, 511, 763, 1017

I

\IeC 26, 387, 637, 891
\if 126, 209, 447, 492, 576, 698, 744, 829, 951, 952, 999
\if@endfor 596, 849, 1033
\if@endpe 216
\if@filesw 34, 897
\if@gls@debug 7, 23, 202, 377, 385, 489, 626, 635, 741, 880, 889, 996
\if@gls@docloaded . 6, 6, 19, 376, 383, 395, 626, 633, 645, 880, 887
\if@glsisacronymlist 22, 123, 385, 445, 634, 696, 889, 949
\if@openright 53, 402, 652, 908
\if@org@KV@glslink@local . . 120, 948
\ifbool 38, 65, 384, 391, 394, 410, 437–439, 633, 641, 644, 660, 688–690, 900, 915
\ifboolexpr 46, 71, 398, 413, 508, 648, 663, 760, 904, 919
\ifcase 7, 9, 10, 30, 36, 39, 41, 78, 228, 320, 377–379, 390, 395, 416, 504, 585, 621, 626, 628, 629, 640, 645, 666, 755, 838, 875, 880, 882, 883, 894, 898, 900, 902, 922, 1011
\ifcsdef 30, 47, 53, 81, 82, 89–94, 104, 203, 218, 219, 221–223, 249, 345, 390, 399, 402, 418, 419, 423–426, 443, 489, 497, 499, 500, 509, 514, 545, 640, 649, 652, 668, 669, 673–677, 694, 741, 749, 751, 752, 761, 766, 798, 894, 905, 908, 924, 925, 929–933, 939, 996, 1004, 1006–1008, 1021, 1033
\ifcsempty 67, 331, 332, 411, 592, 661, 845, 846, 917
\ifcsequal 68, 411, 661, 917

\ifcsname 93, 676, 932
\ifcsstrequal 94, 426, 677, 933
\ifcsstring 93, 425, 676, 932
\ifcsundef 10, 17, 33, 43, 46, 49, 51, 53, 64, 65, 73, 74, 77, 96, 98, 99, 109, 119, 125, 136, 201, 213, 219, 227, 232, 235–238, 241, 248, 346–349, 379, 382, 392, 397, 398, 400–402, 409, 410, 413–415, 427, 429, 434, 444, 446, 453, 454, 482, 483, 488, 494, 496, 497, 503, 506, 508, 509, 511, 514, 546–548, 580, 596, 628, 632, 642, 647, 649, 651, 652, 660, 664, 666, 678, 680, 685, 695, 697, 705, 734, 740, 746, 748, 749, 755, 758, 760, 761, 763, 766, 799–801, 834, 850, 883, 886, 896, 903, 904, 906–908, 915, 920, 921, 934, 936, 942, 948, 950, 951, 959, 995, 1001, 1005, 1010, 1011, 1013–1017, 1021, 1033
\ifcsvoid 69, 917
\ifdef 5, 42, 69, 79, 85, 86, 105, 120, 166, 167, 198, 199, 205, 242, 278, 319, 332, 345, 412, 417, 420, 432, 444, 467, 469, 486, 487, 511, 545, 593, 620, 621, 662, 667, 671, 683, 695, 719, 721, 738, 739, 763, 798, 846, 852, 874, 879, 902, 918, 923, 927, 940, 948, 973, 993, 994, 998, 1018
\ifdefempty 22, 35, 52, 53, 64, 68–70, 74, 114, 116, 194, 196, 224, 225, 247, 248, 254, 263, 350, 352, 367, 385, 393, 402, 409, 411, 412, 414, 437, 440, 441, 484, 485, 501, 502, 513, 514, 519, 524, 528, 530, 532, 548, 551, 553, 564, 635, 643, 652, 659, 661, 662, 664, 665, 688, 691, 692, 736, 737, 753, 754, 765, 766, 771, 777, 780, 782, 785, 801, 804, 806, 817, 889, 898, 908, 915, 917, 918, 920, 921, 945, 946, 990–992, 1008, 1009, 1020, 1021, 1025, 1030
\ifdefequal 34, 67–70, 81, 82, 87, 97, 101, 225, 346, 392, 393, 410–412, 418, 419, 421, 428, 431, 502, 546, 642, 643, 661, 662, 668, 669, 672, 679, 682, 754, 799, 897, 916–918, 924, 925, 928, 935, 938, 1009
\ifdefstrequal 93, 425, 676, 932
\ifdefstring 13, 46, 71, 191–193, 196, 221, 222, 224, 226, 337, 380, 398, 413, 482–484, 499–503, 540, 630, 648, 663, 733–735, 737, 750, 751, 753, 754, 793, 884, 904, 919, 988–990, 992, 1006–1008, 1010
\ifdefvoid . . . 25, 26, 33, 45, 100, 191, 192, 225, 387, 430, 501, 502, 636, 637, 681, 753, 754, 891, 896, 904, 937, 988, 989, 1009
\ifdim 251, 292, 324, 516, 605, 624, 768, 859, 877, 1023
\iffalse 100, 107, 430, 433, 681, 684, 937, 941
\IfFileExists 12, 29, 30, 192–194, 217, 380, 390, 482, 483, 496, 630, 639, 640, 734, 735, 747, 884, 894, 989–991, 1003
\ifglossaryexists 50, 62, 64, 66, 190–193, 401, 408–410, 481–483, 651, 658–660, 733–735, 906, 914–916, 988–990
\ifgls@sanitize@description 27, 388, 638, 892
\ifgls@sanitize@name 27, 388, 638, 892
\ifgls@sanitize@symbol 28, 388, 638, 893
\ifgls@xindy@glsnumbers 63, 409, 659, 914
\ifglsacrdescription 32, 535, 789, 896
\ifglsacrdua 32, 526, 531, 534–536, 779, 784, 787, 789, 896
\ifglsacrfootnote 31, 123, 445, 535, 536, 696, 789, 895, 949
\ifglsacronym . 20, 383, 384, 633, 888
\ifglsacrshortcuts 42, 261, 396, 523, 647, 775, 902, 1030
\ifglsacrsmallcaps 32, 526, 527, 529, 531, 534, 778–780, 782, 784, 786, 787, 896
\ifglsacrsmaller . . 32, 526, 527, 530, 531, 778–780, 782, 784, 896

\ifglsautomake 36, 37, 42, 196, 393, 394, 485, 643, 644, 736, 898, 899, 902, 992
\ifglsdescsuppressed 68, 276, 411, 597, 661, 851, 917
\ifglsentrycounter 14, 15, 42, 230, 231, 381, 397, 505, 506, 630, 631, 647, 757, 758, 884, 885, 902, 1012, 1013
\ifglsentryexists . . . 65, 65, 66, 83–85, 95, 97, 409, 410, 420, 426, 428, 660, 670, 671, 677, 679, 915, 916, 926, 927, 933, 935
\ifglsesclocations 127, 203–205, 490, 742, 952, 997, 998
\ifglsfieldcseq . . . 94, 426, 677, 933
\ifglsfielddefeq . . 93, 425, 676, 932
\ifglsfieldeq 93, 425, 676, 932
\ifglsfieldvoid 68, 917
\ifglshaschildren 67, 276, 410, 581, 598, 660, 834, 851, 916
\ifglshasdesc 67, 276, 411, 598, 661, 851, 917
\ifglshasfield 69, 412, 662, 917
\ifglshaslong 68, 111–113, 168, 249–251, 254, 257, 364, 367, 370, 411, 436, 437, 468, 515, 516, 519, 521, 530, 561, 562, 564, 566, 661, 687, 688, 719, 767, 768, 771, 772, 783, 814, 815, 817, 819, 917, 944, 945, 974, 1021–1023, 1026, 1027
\ifglshasparent 67, 219, 225, 324, 411, 497, 501, 623, 661, 749, 753, 877, 917, 1005, 1009
\ifglshasprefix 331, 332, 334, 335, 592–595, 845–848
\ifglshasprefixfirst 331, 332, 334, 335, 592–595, 846–848
\ifglshasprefixfirstplural 332, 333, 335, 336, 592, 593, 595, 596, 846–849
\ifglshasprefixplural . 331, 333, 334, 336, 592–595, 845, 847–849
\ifglshasshort 68, 212, 411, 662, 917, 1001
\ifglshassymbol 68, 320, 321, 323, 325, 326, 411, 525, 528, 532, 621–625, 661, 777, 780, 785, 875, 876, 878, 917
\ifglshyperfirst . 123, 445, 696, 949
\ifglsindexonlyfirst 203, 489, 741, 997
\ifglsnogroupskip 279, 283, 284, 286, 289, 290, 294, 295, 297, 305, 307, 309, 313, 315, 317, 320, 322, 323, 326, 599, 601, 602, 604, 607, 608, 613–615, 617–619, 621–623, 625, 852, 854–858, 860–862, 867–869, 871–873, 875–877, 879
\ifglsnonumberlist 229, 504, 756, 1012
\ifglsnopostdot . . . 13, 380, 630, 884
\ifglsnumberline . . 54, 403, 653, 908
\ifglsresetcurrcount . . 108, 109, 942
\ifglssanitizesort 25, 28, 386, 389, 636, 638, 639, 890, 893
\ifglssavenumberlist 81, 195, 212, 418, 485, 494, 668, 736, 745, 924, 991, 1001
\ifglssavewrites 38, 190, 203, 394, 481, 489, 644, 733, 741, 899, 988, 996
\ifglssubentrycounter 15, 42, 230, 231, 381, 397, 505, 506, 631, 647, 757, 758, 885, 902, 1012, 1013
\ifglstoc 54, 403, 653, 908
\ifglstranslate 29, 45, 390, 398, 639, 648, 894, 904
\ifglsucmark 51, 52, 401, 651, 652, 907
\ifglsused 65, 111, 114–116, 122, 178, 203, 332–336, 350–352, 410, 436–441, 445, 473, 489, 525, 528, 530, 532, 548–553, 593–595, 660, 686, 688–692, 696, 724, 741, 777, 780, 783, 785, 801–806, 846–849, 915, 943, 945, 946, 949, 979, 997
\ifglswrallowprimitivemods 206, 207, 491, 743, 998
\ifglsxindy 15, 32, 48, 49, 54–63, 74, 101, 102, 127, 179–181, 186, 191, 192, 194, 209, 210, 217, 392, 400, 403–409, 414, 431, 448, 473, 474, 478, 481, 482, 492, 493, 496, 575, 576, 642, 650, 653–659, 664, 682, 699, 725, 726, 730, 733, 734, 743, 745, 748,

828–830, 885, 896, 906, 909–914, 920, 938, 952, 979, 980, 985, 988, 989, 991, 999, 1000, 1003
\ifignoredglossary 75, 96, 100, 203, 213, 415, 427, 430, 489, 494, 665, 678, 681, 741, 746, 921, 934, 937, 996, 1002
\ifin@ 41, 396, 646, 902
\ifinlistcs 222, 227, 500, 503, 752, 755, 1007, 1010
\ifinner 8, 377, 627, 881
\ifKV@glslink@hyper 123, 124, 445, 446, 696, 697, 950
\ifKV@glslink@local 120, 454–457, 706–708, 948
\ifmeasuring@ 103, 432, 683, 939
\ifmmode 8, 377, 627, 881
\ifnum . 16, 109, 110, 193, 224, 292, 321, 323, 325, 326, 382, 393, 435, 484, 501, 585–587, 605, 622–624, 631, 632, 644, 685, 686, 735, 753, 839, 840, 859, 876, 878, 886, 942, 943, 990, 1008
\ifstrempty . 9, 378, 581, 628, 834, 882
\ifstrequal 508, 760
\ifthenelse 29, 43, 53, 125, 195, 389, 397, 402, 446, 484, 508, 536, 596, 639, 647, 652, 697, 736, 760, 789, 849, 893, 903, 908, 950, 991, 1033
\IfTrackedLanguage 187, 478, 730, 985
\IfTrackedLanguageFileExists 46, 328, 329, 399, 574, 591, 649, 827, 844, 905
\iftrue 100, 106, 430, 433, 681, 684, 937, 941
\ifundef . . 14, 15, 72, 84, 96, 181, 185, 194, 381, 413, 420, 427, 474, 477, 484, 630, 631, 663, 670, 678, 726, 729, 736, 884, 885, 919, 926, 934, 980, 983, 991
\ifvmode 178, 472, 724, 979
\ifvoid 292, 606, 859
\ifx 14, 16, 18, 21, 35, 40, 41, 43, 55, 56, 58, 60, 62, 63, 71, 97–99, 102, 104, 125, 128–134, 168, 181, 184, 186, 187, 189, 201, 205, 208, 210, 211, 214, 215, 236–239, 263, 265, 337, 346, 381–385, 396, 397, 403, 404, 406, 408, 409, 427–431, 446, 448–453, 468, 474, 476, 478–480, 488, 490–493, 495, 508–510, 524, 526, 527, 529, 531–533, 535, 537, 539, 545, 576–581, 585–587, 630–632, 634, 643, 646, 647, 653, 654, 656, 657, 659, 678–682, 697, 699–705, 719, 720, 726, 728, 730–732, 740, 742–745, 747, 760–762, 776, 778, 780, 782, 784–786, 788, 790, 793, 799, 830, 832–834, 838–840, 884–889, 898, 901–903, 909, 910, 912, 914, 915, 919, 934–939, 950, 951, 953–958, 974, 980, 983–985, 987, 995, 997–1003, 1015–1017, 1030
\immediate . . 35, 84, 86, 111, 191–194, 201, 217, 393, 420, 421, 436, 481, 483, 489, 496, 643, 670, 671, 686, 687, 733, 735, 740, 741, 748, 898, 926, 927, 943, 944, 988–991, 996, 1003, 1004
\in@ 41, 396, 646, 902
index (option) 40
index (style) 319
indexgroup (style) 320
indexhypergroup (style) 320
\indexname 41, 396, 646, 901
indexonlyfirst (option) 31
\indexspace 278, 279, 299, 299, 300–304, 318, 320–324, 326, 327, 598, 599, 609–613, 620–623, 625, 852, 863–866, 874–877, 879
inline (style) 275
\input 45, 113, 398, 437, 648, 688, 904, 945
\inputencodingname . 33, 392, 642, 896
\InputIfFileExists . 85, 420, 671, 927
\istfile 200, 488, 740
\istfilename 48, 48, 181, 185, 192, 193, 195, 400, 474, 477, 482–484, 576, 579, 650, 726, 729, 734–736, 830, 832, 906, 980, 983, 989–991
\item . 278–281, 299, 300, 319–321, 581, 582, 584, 585, 599, 600, 609, 610, 621, 622, 834, 835, 838, 852–854, 863, 864, 874, 875

J

\jobname 35, 48, 84, 85, 181, 185, 191–194, 216, 217, 393, 400, 420, 474, 477, 481–483, 496, 576, 579, 643, 650, 670, 671, 726, 729, 733–735, 747, 830, 832, 898, 906, 926, 927, 980, 983, 988–991, 1003

K

kernelglossredefs (option) 39
\key@ifundefined 87–89, 422, 672, 673, 928, 929
keys:
access 338
counter 77
description 75
descriptionaccess 339
descriptionplural 75
descriptionpluralaccess 339
first 76
firstaccess 338
firstplural 76
firstpluralaccess 338
long 80
longaccess 339
longplural 80
longpluralaccess 339
name 75
nonumberlist 78
parent 78
plural 76
pluralaccess 338
see 77
short 80
shortaccess 339
shortplural 80
shortpluralaccess 339
sort 76
symbol 76
symbolaccess 338
symbolplural 77
symbolpluralaccess 338
text 76
textaccess 338
type 77
user1 79
user1access 339
user2 79
user2access 339
user3 79
user3access 339
user4 79
user4access 340
user5 79
user5access 340
user6 80
user6access 340
\KV@glslink@hyperfalse 120, 123, 136, 444, 445, 454, 695, 696, 705, 948–950, 959
\KV@glslink@hypertrue 120, 136, 444, 454, 695, 705, 948, 959

L

\L 27, 388, 637, 892
\l 5, 27, 104, 388, 637, 880, 892
\label 11, 103, 104, 228, 230, 231, 379, 504, 505, 629, 756, 757, 883, 939, 940, 1011, 1012
\languagename 33, 392, 642, 896
languages (option) 31
\leaders 281, 582, 600, 835, 853
\leavevmode 94, 123, 426, 445, 677, 696, 933, 950
\let 6, 7, 13, 16–18, 20, 26, 27, 29–31, 33, 35–43, 46, 47, 57, 69–71, 81, 83, 94, 95, 97, 99, 100, 103–108, 110, 113, 120–124, 127, 135–143, 158–165, 168, 176, 185, 186, 190–193, 195, 197–199, 202, 204, 205, 207, 211–216, 225, 227, 230, 247, 259–261, 274, 278, 292, 299, 300, 319, 328, 330, 340, 341, 355–357, 376, 377, 380, 382–384, 387, 388, 390, 391, 393–398, 400, 404, 405, 412, 413, 418, 419, 426–430, 432–435, 437, 444–448, 453–457, 463–468, 471, 472, 477, 478, 481–487, 489–491, 493–495, 502, 503, 505, 507, 513, 523–535, 542, 555, 556, 569–574, 580, 591, 592, 596, 597, 605, 609, 610, 621, 626, 627, 630–633, 637–641, 643–650, 655, 662, 663, 668–670, 677–681, 683–686, 688, 695–699, 705–709, 714–720,

723, 724, 728, 730, 733–739, 741–743, 745–747, 754, 755, 757, 759, 765, 775, 776, 778–786, 788, 795, 796, 808–810, 822–827, 833, 844, 845, 849, 850, 852, 859, 863, 864, 874, 880, 881, 884–888, 891, 892, 894–905, 910, 911, 918, 919, 924–926, 933–937, 939–943, 945, 948–950, 952, 959–962, 968–974, 978, 983, 984, 988–998, 1001–1003, 1009–1012, 1020, 1029, 1030, 1033, 1034
\letcs . . 33, 35, 67–70, 86, 88, 89, 92, 98, 99, 166, 168, 198, 199, 218, 220, 224, 225, 233, 235, 236, 324, 410–412, 420, 422, 425, 429, 467, 468, 482, 483, 486, 487, 497, 498, 501, 502, 506, 508, 624, 643, 661, 662, 671, 673, 676, 679, 680, 719, 734, 735, 738, 739, 749, 750, 753, 754, 758, 760, 877, 896, 898, 916–918, 927, 929, 932, 935, 936, 973, 974, 993, 994, 1004–1006, 1009, 1013–1015, 1033
list (style) 278
\listcsadd 222, 500, 752, 1007
\listcsgadd 227, 503, 755, 1010
\listcsxadd 218, 497, 749, 1004
listdotted (style) 281
\listeadd 222, 500, 752, 1007
listgroup (style) 279
listhypergroup (style) 279
\listxadd 84, 670, 926
\loadglsentries 113, 113, 437, 688, 945
\long 94, 240, 426, 511, 677, 762, 933, 1017
long (key) 80
long (style) 282
long-booktabs (style) 288
long-sc-short (acrstyle) 251
long-sc-short-desc (acrstyle) 253, 365
long-short (acrstyle) 249, 364
long-short-desc (acrstyle) . . 252, 365
long-sm-short (acrstyle) 252
long-sm-short-desc (acrstyle) 253, 366
long-sp-short (acrstyle) 250
long-sp-short-desc (acrstyle) . . . 253
long3col (style) 283
long3col-booktabs (style) 289
long3colborder (style) 284
long3colheader (style) 285
long3colheaderborder (style) 285
long4col (style) 285
long4col-booktabs (style) 289
long4colborder (style) 286
long4colheader (style) 286
long4colheaderborder (style) 286
longaccess (key) 339
longborder (style) 283
longheader (style) 283
longheaderborder (style) 283
\longnewglossaryentry 94, 95, 426, 677, 933
longplural (key) 80
longpluralaccess (key) 339
\longprovideglossaryentry 95, 426, 677, 933
longragged (style) 293
longragged-booktabs (style) 290
longragged3col (style) 295
longragged3col-booktabs (style) . 291
longragged3colborder (style) 296
longragged3colheader (style) 296
longragged3colheaderborder (style) 296
longraggedborder (style) 294
longraggedheader (style) 294
longraggedheaderborder (style) . . 295
\LT@end@pen 292, 605, 859
\LT@err 292, 605, 859
\LT@foot 292, 605, 606, 859
\LT@head 292, 605, 606, 859
\LT@lastfoot 292, 605, 606, 859
\LT@output 292, 605, 859

M

\makeatletter 85, 216, 420, 496, 671, 747, 927, 1003
\makebox 281, 324–326, 582, 586, 587, 600, 624, 625, 835, 840, 853, 877, 878
\makefirstuc 4, 879
\makeglossaries 9, 34–36, 39, 40, 44, 78, 191, 194, 196, 198, 200, 217, 378, 392–395, 398, 416, 481, 484–486, 488, 496, 628, 642–645, 648, 666, 733, 736–738, 740,

748, 882, 897, 898, 900, 901, 903, 922, 988, 991–993, 995, 1004
\makeglossary 38, 38, 39, 40, 394, 395, 644, 645, 900, 901
makeindex (option) 33
\makenoidxglossaries 9, 34, 35, 78, 195, 196, 200, 378, 392, 393, 416, 485–488, 628, 642, 643, 666, 736–740, 882, 897, 922, 992–995
\MakeTextUppercase . . 5, 376, 626, 880
\MakeUppercase 549, 550, 556, 802–804, 809, 810
\marginpar 8, 378, 627, 881
\markboth 51, 52, 401, 651, 907
\mbox 178, 280, 303, 304, 325, 472, 582, 599, 612, 624, 724, 835, 853, 865, 866, 878, 979
mcolalttree (style) 303
mcolalttreegroup (style) 303
mcolalttreehypergroup (style) . . . 303
mcolalttreespannav (style) 304
mcolindex (style) 299
mcolindexgroup (style) 299
mcolindexhypergroup (style) 299
mcolindexspannav (style) 300
mcoltree (style) 300
mcoltreegroup (style) 301
mcoltreehypergroup (style) 301
mcoltreenoname (style) 301
mcoltreenonamegroup (style) 302
mcoltreenonamehypergroup (style) 302
mcoltreenonamespannav (style) . . . 302
mcoltreespannav (style) 301
\memUChead 51, 401, 651, 907
\MessageBreak . . . 23, 39, 71, 214, 215, 328, 329, 337, 386, 394, 395, 413, 495, 539, 574, 591, 635, 644, 645, 663, 747, 793, 827, 828, 844, 890, 900, 919, 1002
mfirstuc (option) 41
\mfirstucMakeUppercase 5, 376, 401, 424, 438–442, 457–467, 514, 519–521, 528, 532, 552–554, 557–560, 564, 565, 567, 595, 596, 626, 651, 652, 674, 689–693, 709–717, 719, 765, 766, 771–773, 781, 785, 805–807, 810–813, 817, 818, 820, 848, 849, 880
\MFUaddmap 5, 879
\MFUblocker 5, 879
\MFUexcl 5, 879
\MFUsentencecase 5, 5, 41, 167, 171, 879, 880, 902, 973, 976
\midrule 289, 290, 604, 857, 858
\month . . . 181, 185, 474, 477, 576, 579, 726, 729, 830, 832, 980, 983

N

\n 185, 477, 579, 580, 729, 832, 833, 983, 984
name (key) 75
\NeedsTeXFormat 4, 271, 275, 277, 281, 288, 293, 298, 304, 311, 318, 327–329, 336, 375, 376, 539, 574, 575, 580, 591, 626, 792, 827, 828, 834, 844, 852, 879, 1033
\new@glossaryentry . . 83, 84, 198, 419, 420, 486, 670, 738, 926, 993
\new@ifnextchar 72, 89, 90, 111–113, 137–141, 143–165, 244–246, 332–336, 413, 423, 424, 436, 454–467, 511, 512, 593–595, 663, 674, 687, 706–718, 763, 764, 846–848, 919, 930, 944, 959–972, 1018, 1019
\newacronym 241, 242, 242, 247, 511, 513, 524, 526, 527, 529, 531, 533, 535, 537, 763, 765, 776, 778, 780, 782, 784, 786, 788, 790, 1017, 1018, 1020
\newacronymhook 246, 247, 373, 513, 524, 526, 528, 529, 531, 534, 535, 537, 569, 765, 776, 778, 780, 782, 784, 787, 788, 790, 822, 1020
\newacronymstyle 249, 249, 250–254, 256, 258, 259, 514–520, 522, 766–772, 774, 1021–1025, 1027–1029
\newcommand 4, 5, 8–33, 35–37, 39–45, 47–90, 93–95, 100–114, 116, 118–125, 127, 134–180, 186, 187, 189–194, 196, 200–205, 207, 209–214, 216, 218–224, 226–238, 240–246, 248, 249, 251, 259, 261–263, 265–278, 292,

299, 318, 319, 324, 330–336, 338, 342–346, 363, 373–375, 377–387, 389–411, 413–426, 430–437, 440, 441, 443–448, 453–474, 478, 481–485, 487–494, 496–509, 511–516, 523–540, 544–546, 561, 569, 573, 574, 580, 592–599, 605, 609, 620, 621, 624, 627–637, 639–641, 643–648, 650–655, 657–661, 663–677, 681–688, 691–699, 705–725, 730, 733–737, 739–768, 775–793, 797–799, 814, 822, 826, 827, 834, 845–852, 859, 863, 874, 877, 879–930, 932, 933, 937–946, 948–950, 952, 959–980, 985, 987–992, 995–1021, 1023, 1029–1034
\newcount 17, 81, 382, 418, 632, 668, 886, 924
\newcounter 14, 15, 381, 630, 631, 884, 885
\NewDocumentCommand 134
\newenvironment . 232, 506, 758, 1013
\newglossary 19, 20, 40, 41, 72, 73, 195, 383, 384, 395, 396, 413, 414, 484, 633, 634, 646, 663, 664, 736, 887, 888, 901, 919, 920, 991
\newglossaryentry 9, 41, 80, 83, 83, 108, 247, 378, 396, 417, 419, 434, 513, 523, 525, 526, 528, 530, 532, 534, 536, 569–572, 628, 646, 667–670, 685, 765, 775, 777, 779, 781, 783, 785, 787, 789, 822–825, 882, 901, 923–926, 942, 1020
\newglossarystyle 237, 275, 278–291, 293–317, 319–324, 327, 509, 597, 599–625, 761, 850, 852–858, 860–877, 879, 1015
\newif 6, 7, 22, 29, 32, 36, 108, 206, 376, 377, 385, 390, 392, 393, 491, 626, 634, 639, 642, 643, 743, 880, 889, 894, 896, 898, 942, 998
\newignoredglossary 74, 414, 664, 920
\newlabel 135
\newlength . . . 134, 281, 282, 293, 305, 312, 322, 453, 600, 606, 613, 617, 622, 705, 853, 854, 860, 866, 867, 870, 871, 876, 959
\newrobustcmd 8, 9, 67–69, 83, 84, 89–93, 111, 112, 122, 137–166, 168, 171, 177, 178, 211, 212, 235, 238, 242–246, 263, 324, 332–335, 346–349, 378, 410–412, 419, 420, 423, 424, 436, 444, 454–473, 493, 494, 511, 512, 546–548, 592–595, 623, 627, 628, 660–662, 669, 670, 674, 687, 695, 706–724, 745, 763, 764, 799–801, 845–848, 877, 882, 916, 917, 925, 926, 930–932, 944, 949, 959–974, 976, 979, 1001, 1014, 1016, 1018, 1019, 1030
\newterm 41, 396, 646, 901
\newtoks 128, 190, 246, 448, 481, 513, 699, 733, 765, 953, 988, 1019, 1020
\newwrite . . 35, 84, 181, 185, 190, 194, 393, 420, 474, 477, 481, 484, 643, 670, 726, 729, 733, 736, 898, 926, 980, 983, 988, 991
\nfss@text 8, 377, 627, 881
\noalign 292, 605, 859
\nobreak 280, 292, 582, 599, 606, 835, 853, 859
\noexpand 21, 43, 55, 56, 85, 99, 100, 104, 125–128, 133, 134, 176, 187–189, 191–193, 195, 204, 209, 212, 217, 220, 240, 241, 247, 384, 397, 403, 404, 420, 429, 430, 443, 444, 446–448, 453, 472, 478–483, 485, 490, 491, 494, 496, 498, 506, 507, 511, 513, 523, 525–530, 532–534, 536, 540, 569–573, 575, 634, 648, 654, 671, 680, 681, 694, 697–699, 704, 705, 723, 730–736, 742, 743, 746, 748, 750, 758, 759, 763, 765, 775, 777–779, 781, 783, 785–787, 789, 790, 793, 822–826, 828, 888, 903, 909, 910, 927, 936, 937, 939, 950–953, 958, 978, 985–992, 997, 1000, 1001, 1003–1006, 1017, 1020
noglossaryindex (option) 41
nogroupskip (option) 14

\nohyperpage . 238, 510, 761, 762, 1016
nohypertypes (option) 23
\noindent 300–304, 322–324, 507, 610–612, 622, 623, 759, 864–866, 876, 877
\noist 185, 186, 190, 190, 481, 579, 580, 733, 832, 833, 983, 984, 987, 988
nolangwarn (option) 23
nolist (option) 13
nolong (option) 12
nomain (option) 20
nonumberlist (key) 78
nonumberlist (option) 12
\nopostdesc 41, 47, 47, 94, 215, 396, 399, 400, 426, 495, 581, 646, 650, 677, 747, 834, 901, 905, 933, 1003
nopostdot (option) 13
noredefwarn (option) 24
\normalbaselineskip . . . 292, 605, 859
nostyles (option) 13
nosuper (option) 13
notranslate (option) 30
notree (option) 13
nowarn (option) 23
\ns@ACRfull 245, 512, 764, 1019
\ns@Acrfull 244, 512, 763, 1019
\ns@acrfull 244, 511, 763, 1018
\ns@ACRfullpl . . . 246, 512, 764, 1019
\ns@Acrfullpl . . . 245, 512, 764, 1019
\ns@acrfullpl . . . 245, 512, 764, 1019
\ns@ACRlong 163, 466, 717, 971
\ns@Acrlong 162, 465, 717, 971
\ns@acrlong . . . 161, 162, 465, 716, 970
\ns@ACRlongpl 165, 467, 718, 972
\ns@Acrlongpl 164, 466, 718, 972
\ns@acrlongpl 164, 466, 718, 972
\ns@ACRshort 159, 463, 715, 969
\ns@Acrshort 158, 463, 715, 968
\ns@acrshort 158, 463, 714, 968
\ns@ACRshortpl . . . 161, 464, 716, 970
\ns@Acrshortpl . . . 160, 464, 716, 970
\ns@acrshortpl . . . 160, 464, 715, 969
\ns@newglossary 72, 72, 413, 663, 919
\null 127–134, 187–189, 217, 447–453, 478–481, 496, 698–705, 730–732, 748, 952–958, 985–987, 1003
\number . 16, 85, 98, 104, 110, 204, 207, 208, 382, 420, 428, 435, 490, 491, 507, 540, 632, 671, 679, 686, 742, 743, 759, 793, 886, 927, 935, 939, 943, 997, 998
numberedsection (option) 10
\numberline 54, 403, 653, 909
numberline (option) 10
numbers (option) 40
\numexpr 110, 435, 686, 943

O

\O 27, 388, 637, 892
\o 27, 388, 638, 892
\OE 27, 388, 637, 892
\oe 27, 388, 637, 892
\oldacronym 241, 511, 763, 1017
\openout . . . 35, 84, 181, 185, 191, 393, 420, 474, 477, 481, 576, 579, 643, 670, 726, 729, 733, 830, 832, 898, 926, 980, 983, 988
options:
acronym 20
acronymlists 22
acronyms 20
automake 36
automakegloss 37
automakeglosslite 37
compatible-2.07 38
compatible-3.07 38
counter 22
counterwithin 14
debug 7
description 32
disablemakegloss 34
dua 32
entrycounter 14
esclocations 12
footnote 31
hyperfirst 31
index 40
indexonlyfirst 31
kernelglossredefs 39
languages 31
makeindex 33
mfirstuc 41
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 32
restoremakegloss 34
sanitize 29
sanitizesort 28
savenumberlist 12
savewrites 38
section 10
seeautonumberlist 12
seenoindex 9
shortcuts 32
smallcaps 32
smaller 32
sort 15
style 11
subentrycounter 15
symbols 40
toc 10
translate 30
ucmark 14
writeglslabelnames 36
writeglslabels 36
xindy 33
xindygloss 34
xindynoglsnumbers 34
\OR 536, 789
\or 7, 9–11, 30, 36, 37, 39–41, 228, 320, 377, 379, 391, 395, 504, 585, 621, 626–629, 640, 645, 755, 756, 838, 875, 881–883, 895, 899–902, 1011
order (option) 32
\org@glossaryentrynumbers . 215, 230, 495, 505, 747, 757, 1003, 1012
\org@glossarytitle 214, 215, 495, 746, 747, 1002, 1003
\org@glspostdescription 47, 400, 650, 905
\org@ifKV@glslink@hyper 124, 445, 446, 696, 697, 950
\outputpenalty 292, 605, 859

P

\p@ 278, 299, 318, 319, 599, 609, 620, 621, 852, 863, 874
\p@gls@hyp@opt 121, 121, 444, 695, 949
\PackageError 9, 18, 19, 27, 36, 44, 55, 62, 65, 66, 71, 77, 80, 81, 88–94, 96–98, 108, 120, 166, 189, 191, 195, 198, 200, 221–223, 227, 229, 236, 237, 242, 248, 249, 349, 375, 376, 378, 383, 394, 398, 404, 408, 410, 413, 415, 417, 418, 422–428, 434, 444, 467, 481, 484, 486–488, 499, 500, 503, 504, 508, 509, 514, 515, 526, 527, 531, 534, 548, 581, 628, 633, 644, 648, 654, 658, 660, 663, 666–668, 673–679, 685, 695, 719, 733, 736, 738–740, 751, 752, 755, 756, 760, 761, 766, 767, 779, 780, 784, 787, 801, 834, 882, 887, 892, 898, 903, 910, 914, 916, 919, 921, 923, 924, 929–935, 942, 948, 973, 987, 988, 991, 993, 995, 1006–1008, 1011, 1012, 1015, 1016, 1018, 1021
\PackageInfo 7, 34, 35, 191, 202, 377, 393, 481, 489, 626, 627, 642, 643, 733, 741, 881, 897, 988, 996
\PackageWarning 7, 23, 377, 385, 626, 627, 635, 880, 881, 889
\PackageWarningNoLine 7, 23, 328, 329, 377, 385, 386, 574, 591, 626, 627, 635, 827, 844, 880, 881, 889, 890
\pagegoal 292, 605, 859
\pagelistname . . . 44, 47, 285–287, 289, 290, 296, 298, 308–311, 315–318, 398, 399, 602–604, 608, 609, 615–617, 619, 620, 648, 649, 855, 856, 858, 861–863, 868–870, 872–874, 903, 905
\par 47, 278, 280, 299, 301–304, 318, 319, 321–

327, 400, 507, 508, 582, 585–587, 599, 609–613, 620–625, 650, 759, 760, 835, 839, 840, 852, 853, 863–866, 874–879, 905
parent (key) 78
\parindent 299–304, 319, 321–323, 325–327, 585–587, 609–613, 621–625, 838–840, 863–866, 874–876, 878, 879
\parskip 299–302, 319, 321, 322, 609–611, 621–623, 863–865, 874–876
\PassOptionsToPackage 329, 336, 539, 591, 792, 844
\pdfstringdefPreHook 205, 998
\penalty 292, 605, 859
\PGLS 335, 595, 848
\Pgls 333, 333, 594, 847
\pgls 332, 333, 593, 846
\PGLSpl 335, 595, 848
\Pglspl 334, 334, 594, 847
\pglspl 333, 334, 593, 846
\phantomsection . . . 53, 402, 652, 908
plural (key) 76
pluralaccess (key) 338
\printacronyms 20, 384, 633, 888
\PrintChanges 6, 376, 626, 880
\printglossaries 196, 214, 485, 494, 737, 746, 992, 1002
\printglossary . . . 20, 24, 40, 41, 196, 213, 214, 229, 384, 386, 395, 396, 485, 494, 504, 634, 636, 646, 737, 746, 756, 888, 890, 901, 992, 1001, 1002, 1012
\printindex 41, 396, 646, 901
\printnoidxglossaries 197, 214, 486, 494, 738, 746, 993, 1002
\printnoidxglossary 197, 200, 214, 214, 221, 222, 229, 485, 486, 488, 494, 499, 500, 504, 737, 738, 740, 746, 751, 752, 756, 992, 993, 995, 1002, 1006, 1007, 1012
\printnumbers 40, 396, 646, 901
\printsymbols 40, 395, 646, 901
\ProcessOptions 329, 336, 539, 591, 792, 844
\ProcessOptionsX . . 42, 396, 646, 902
\protect 54, 118, 119, 167, 249–251, 257, 273, 354, 355, 364, 365, 370, 403, 443, 515, 516, 521, 525, 528, 530, 555, 561, 562, 566, 653, 693, 694, 767, 768, 773, 777, 781, 783, 808, 814, 815, 819, 909, 948, 973, 1022, 1023, 1027, 1028, 1033
\protected@csedef . 91, 424, 675, 931
\protected@csxappto 1033
\protected@csxdef 91, 424, 675, 931, 1033
\protected@eappto 35, 74, 75, 101, 204, 643, 664, 665, 682, 742, 898, 920, 921, 938, 997
\protected@edef 8, 11, 18, 21, 35, 56, 58, 61, 63, 64, 67, 71, 72, 74, 75, 91–97, 100–102, 118, 119, 123–126, 168, 176–178, 194, 196, 197, 203, 206, 208, 209, 212, 218, 219, 228, 231, 237, 239, 247, 263, 271, 345, 346, 373, 378, 379, 404, 405, 407, 409, 428, 430, 431, 437–439, 443, 446, 447, 472, 489–491, 504, 506, 507, 509, 513, 532, 536, 540, 545, 546, 576, 577, 592, 596, 627, 629, 632, 634, 643, 654, 656, 658, 659, 661, 663–665, 675–679, 681, 682, 688–690, 694, 696–698, 723, 724, 737, 741–743, 746, 748, 749, 756–759, 761, 765, 785, 789, 793, 798, 799, 822, 829, 830, 845, 849, 882, 883, 886–888, 898, 910, 911, 913, 915, 916, 919–921, 931–935, 937, 938, 948, 950, 951, 974, 978, 979, 990, 992, 993, 996, 998, 999, 1001, 1004, 1005, 1011, 1012, 1015, 1017, 1020, 1030, 1033
\protected@write 71, 73, 135, 181–183, 194, 195, 197, 200, 203, 212, 273, 413, 474, 475, 484, 485, 488, 489, 494, 577, 596, 663, 664, 726, 727, 736, 737, 740, 741, 746, 830, 849, 919, 920, 981, 982, 991, 993, 995, 996, 1001, 1033
\protected@xdef 16–18, 21, 26, 82, 97,

100, 102, 103, 208, 216, 341, 342, 382–384, 387, 418, 431, 432, 491, 542–544, 631, 632, 634, 637, 669, 679, 681–683, 743, 747, 796, 797, 886–888, 891, 925, 935, 937–939, 999, 1003
\providecommand 5, 20, 34, 44, 45, 52, 71, 111, 136, 183, 194, 197, 217, 278, 299, 318, 332, 337, 384, 392, 398, 401, 413, 436, 454, 475, 484, 486, 488, 496, 506, 507, 540, 593, 598, 609, 620, 633, 642, 648, 652, 663, 686, 706, 727, 736, 738, 740, 748, 758, 759, 793, 846, 852, 863, 874, 880, 888, 897, 903, 904, 907, 919, 943, 959, 982, 991, 993, 1003, 1004
\provideglossaryentry 83, 420, 670, 926
\ProvidesFile 45, 398, 648, 904
\ProvidesGlossariesLang 45, 398, 648, 904
\ProvidesPackage 4, 271, 275, 277, 282, 288, 293, 299, 304, 312, 318, 328, 329, 336, 375, 376, 539, 574, 575, 580, 591, 596–598, 600, 604, 606, 609, 613, 617, 620, 626, 792, 827, 828, 834, 844, 849, 850, 852, 854, 857, 860, 863, 866, 870, 874, 879, 1033

R

\r 26, 387, 637, 891
\raggedright 291, 293–298, 312–318, 605–609, 617–620, 858–863, 871–874
\raisebox 134, 453, 705, 959
\ref 231, 505, 757, 1013
\refstepcounter 103, 104, 230, 231, 505, 757, 939, 940, 1012
\regex 235, 1014
\relax 7, 10, 13, 17, 18, 20, 30, 31, 35, 36, 39, 41, 42, 57, 71, 76, 78, 82, 85, 97, 100, 105, 109, 110, 121, 122, 125–134, 168, 183–187, 189, 190, 193, 195, 197–199, 204, 208, 209, 211, 214, 216, 224, 225, 227, 228, 235, 237, 265, 274, 278, 292, 299, 303, 304, 318, 320, 321, 323–327, 340, 341, 355–357, 377, 379, 380, 382–384, 390, 391, 393, 395–397, 405, 413, 415, 416, 419, 420, 428, 430, 432, 435, 444–453, 468, 476–481, 484–487, 490–493, 495, 501, 503, 504, 508, 509, 537, 542, 555, 556, 569–572, 576, 578–580, 585–587, 596, 599, 605, 609, 612, 613, 620–626, 629, 630, 632, 633, 640, 643–647, 655, 663, 665, 666, 669, 671, 679, 681, 683, 685, 686, 695, 697–704, 719, 720, 728–730, 732, 733, 735–739, 742–745, 747, 753, 755, 760, 761, 790, 795, 796, 808, 809, 822–824, 826, 829, 831–833, 838–840, 849, 852, 859, 863, 866, 874–880, 883, 884, 886–888, 894, 895, 898, 900–903, 911, 919, 921, 922, 925, 927, 935, 937, 940, 942, 943, 949, 951–958, 974, 982–985, 987, 988, 990, 991, 993, 994, 997, 999–1003, 1008–1011, 1015, 1030, 1033
\renewacronymstyle . 249, 364–367, 369–372, 514, 561–564, 566–568, 766, 814–817, 819, 821, 1021
\renewcommand 5–7, 9–15, 18–20, 22–24, 28, 30, 31, 34, 36–38, 40–42, 46–49, 62, 75, 77, 79, 94, 104, 108–110, 176, 178, 180, 186, 188, 194–199, 216, 217, 228, 242, 247–259, 275, 276, 279–287, 289, 290, 292–309, 312–317, 319–327, 337, 341, 350, 352, 354, 355, 362–373, 376–381, 383–386, 389–396, 398–400, 408, 414, 416, 426, 434, 435, 472–474, 478, 479, 484–487, 495, 496, 504, 513–524, 526, 527, 529–531, 533–535, 537, 540, 542, 548, 551, 553–555, 560–572, 575, 576, 580–590, 597–615, 617–619, 621–631, 633–636, 638–642, 644, 645,

647, 649, 650, 658, 665–667, 677, 685, 686, 723–725, 730, 731, 736–739, 747, 748, 755, 756, 765–780, 782, 784, 786–788, 790, 793, 795, 796, 801, 804, 806, 808, 813–825, 828, 829, 833–843, 850–869, 871–885, 887–890, 893–895, 897, 899, 901, 902, 904–906, 914, 921–923, 933, 940, 942, 943, 978–980, 985, 986, 991–994, 1003, 1004, 1011, 1018, 1020–1029
\renewenvironment 232, 275, 279, 282–288, 290, 291, 293–319, 321, 322, 325, 506, 597, 599–603, 605–624, 758, 850, 852, 854–876, 878, 1013
\renewglossarystyle 237, 509, 761, 1015
\RequireGlossariesLang 45, 46, 328, 329, 398, 399, 574, 591, 648, 649, 827, 844, 904, 905
\RequirePackage 4–6, 12, 13, 29, 30, 42, 45, 264, 282, 288, 293, 299, 304, 312, 328, 329, 337, 376, 380, 390, 396, 398, 537, 539, 540, 574, 591, 600, 604, 606, 609, 613, 617, 626, 629, 630, 639, 640, 646, 648, 790, 792, 793, 827, 844, 854, 857, 860, 863, 866, 870, 879, 880, 884, 894, 895, 902, 904, 1030
\restorecounters@ 125, 446, 697, 951
restoremakegloss (option) 34
\romannumeral . 204, 207, 208, 324, 326, 490, 491, 587, 624, 625, 742, 743, 840, 877, 878, 997, 998

S

\s@gls@hyp@opt 121, 121, 444, 695, 949
\s@GlsSetXdyFirstLetterAfterDigits 179, 180, 473, 725, 979, 980
\s@GlsSetXdyNumberGroupOrder 180, 473, 474, 725, 980
\s@ifglossaryexists . 64, 64, 213, 409, 494, 659, 660, 746, 915, 1002
\s@newglossary . 72, 72, 413, 663, 919
sanitize (option) 29
sanitizesort (option) 28
\savecounters@ . . . 125, 446, 697, 951
savenumberlist (option) 12
savewrites (option) 38
sc-short-long (acrstyle) 252
sc-short-long-desc (acrstyle) 254, 366
section (option) 10
see (key) 77
seeautonumberlist (option) 12
\seename 45, 211, 398, 493, 648, 745, 904, 1001
seenoindex (option) 9
\seq 272, 274
\setabbreviationstyle 264, 788, 1030
\SetAcronymLists . . 22, 385, 634, 889
\SetAcronymStyle 264, 391, 392, 535, 536, 641, 788, 789
\setacronymstyle . 242, 248, 264, 514, 766, 788, 1018, 1021, 1030
\setbool 28, 389, 639, 893
\setbox 292, 605, 606, 859
\setcounter 104, 230, 505, 757, 939, 1012
\SetCustomDisplayStyle 264, 536, 537, 789, 790
\SetCustomStyle 264, 537, 790
\SetDefaultAcronymDisplayStyle . . . 261, 523, 524, 775, 776, 1030
\SetDefaultAcronymStyle 261, 524, 535, 776, 788
\SetDescriptionAcronymDisplayStyle 262, 528, 529, 780, 782
\SetDescriptionAcronymStyle 262, 529, 535, 782, 789
\SetDescriptionDUAAcronymDisplayStyle 262, 526–528, 779, 780
\SetDescriptionDUAAcronymStyle 262, 527, 535, 780, 789
\SetDescriptionFootnoteAcronymDisplayStyle 262, 524, 526, 777–779
\SetDescriptionFootnoteAcronymStyle 262, 526, 535, 778, 789
\SetDUADisplayStyle 263, 534, 535, 787, 788
\SetDUAStyle . . 263, 535, 536, 788, 789
\setentrycounter 56, 183, 227, 236, 404, 475, 503, 508, 575, 654, 727, 755, 760, 828, 910, 982, 1010, 1015
\SetFootnoteAcronymDisplayStyle 262, 530, 531, 782, 784

\SetFootnoteAcronymStyle 263, 531, 536, 784, 789
\SetGenericNewAcronym 246, 248, 513, 514, 765, 766, 1020, 1021
\setglossarypreamble 50, 401, 651, 906
\setglossarysection 52, 402, 652, 907
\setglossarystyle 215, 228, 236, 265, 279–281, 283–291, 294–304, 306–311, 313–315, 317, 320, 322, 323, 327, 495, 508, 509, 537, 599–605, 607–623, 625, 747, 760, 761, 790, 852–877, 879, 1002, 1011, 1015, 1031
\setglossentrycompatibility . . . 234, 503, 507, 509, 755, 759, 761
\setkeys 29, 33, 42, 52, 96, 124, 177, 178, 215, 247, 389, 392, 397, 402, 427, 445, 472, 473, 495, 513, 524, 526, 527, 529, 531, 533, 535, 537, 639, 642, 647, 652, 678, 696, 724, 747, 765, 776, 778, 780, 782, 784, 786, 788, 790, 894, 897, 903, 907, 934, 950, 979, 1002, 1020
\setlength 281, 282, 293, 299–302, 305, 312, 319, 321, 322, 326, 587, 600, 606, 609–611, 613, 617, 621–625, 840, 853, 854, 860, 863–867, 870, 871, 874–876, 878
\SetSmallAcronymDisplayStyle . . 263, 532–534, 785–787
\SetSmallAcronymStyle 263, 533, 536, 786, 789
\setStyleFile . 47, 400, 650, 905, 906
\settodepth 105, 940
\settoheight 105, 453, 705, 940
\settowidth 105, 251, 516, 586, 587, 624, 625, 768, 840, 877, 878, 940, 1023
\setupglossaries . . 42, 396, 647, 902
\sfcode 13, 380, 630, 884
short (key) 80
short-long (acrstyle) 251, 364
short-long-desc (acrstyle) . . 253, 366
shortaccess (key) 339
shortcuts (option) 32
shortplural (key) 80
shortpluralaccess (key) 339
\show 265–270, 374, 375, 537–539, 573, 574, 790–792, 826, 827, 1031, 1032
\showacronymlists 269, 539, 792, 1032
\showglocounter . 266, 537, 791, 1031
\showglodesc 267, 538, 791, 1031
\showglodescaccess 374, 574, 827
\showglodescplural 268, 538, 791, 1032
\showglodescpluralaccess 374, 574, 827
\showglofield . . . 269, 539, 792, 1032
\showglofirst . . . 265, 537, 791, 1031
\showglofirstaccess . . . 374, 573, 826
\showglofirstpl . 266, 537, 791, 1031
\showglofirstpluralaccess 374, 573, 827
\showgloflag 269, 538, 792, 1032
\showgloindex . . . 269, 538, 792, 1032
\showglolevel . . . 265, 537, 790, 1031
\showgloloclist . 269, 539, 792, 1032
\showglolong 268, 538, 792, 1032
\showglolongaccess 375, 574, 827
\showglolongpluralaccess 375, 574, 827
\showgloname 267, 538, 791, 1031
\showglonameaccess 374, 573, 826
\showgloparent . . 265, 537, 790, 1031
\showgloplural . . 265, 537, 790, 1031
\showglopluralaccess . . 374, 573, 826
\showgloshort . . . 268, 538, 791, 1032
\showgloshortaccess . . . 375, 574, 827
\showgloshortpluralaccess 375, 574, 827
\showglosort 268, 538, 791, 1032
\showglossaries . 269, 539, 792, 1032
\showglossarycounter 270, 539, 792, 1032
\showglossaryentries 270, 539, 792, 1032
\showglossaryin . 270, 539, 792, 1032
\showglossaryout 270, 539, 792, 1032
\showglossarytitle 270, 539, 792, 1032
\showglosymbol . . 268, 538, 791, 1032
\showglosymbolaccess . . 374, 573, 827
\showglosymbolplural 268, 538, 791, 1032
\showglosymbolpluralaccess 374, 573, 827
\showglotext 265, 537, 790, 1031
\showglotextaccess 374, 573, 826

\showglotype 266, 537, 791, 1031
\showglouseri ... 266, 538, 791, 1031
\showglouserii .. 266, 538, 791, 1031
\showglouseriii . 267, 538, 791, 1031
\showglouseriv .. 267, 538, 791, 1031
\showglouserv ... 267, 538, 791, 1031
\showglouservi .. 267, 538, 791, 1031
sm-short-long (acrstyle) 252
sm-short-long-desc (acrstyle) 254, 367
smallcaps (option) 32
smaller (option) 32
\SmallNewAcronymDef 263, 373, 532, 534, 572, 785, 787, 825
sort (key) 76
sort (option) 15
\space ... 9, 34–36, 39, 44, 55, 56, 60, 63, 71, 78, 80, 108, 111–113, 118, 119, 175, 181–185, 189, 191–197, 200, 205, 212, 215, 217, 231, 243, 249–259, 263, 274, 277, 279, 280, 282, 294, 305, 312, 313, 319–321, 323, 325, 326, 354, 355, 364–373, 378, 392–395, 398, 404, 407, 409, 416, 417, 434, 436, 437, 443, 444, 471, 474–477, 481–488, 493, 495, 496, 505, 506, 509, 512, 515–523, 528, 532, 554, 555, 561–569, 575–579, 581–589, 597–601, 606, 613, 617, 621–625, 628, 642–645, 648, 654, 657, 659, 666–668, 685, 687, 693–695, 723, 726–729, 733–740, 745, 747, 748, 757, 761, 763, 767–775, 781, 785, 808, 814–822, 828–832, 835, 837–843, 850–854, 860, 867, 871, 874–878, 882, 897, 898, 900, 903, 910, 912–915, 919, 922–924, 942, 944, 945, 948, 978, 981–983, 987–993, 995, 997, 1001, 1002, 1004, 1013, 1018, 1022–1030, 1034
\spacefactor 13, 380, 630, 884
\SS 27, 388, 638, 892
\ss 27, 388, 638, 892
\str 236, 1015
\string 9, 18, 19, 24, 27, 34–36, 39, 44, 54–56, 58–61, 63, 71, 73, 78, 80, 86, 89, 90, 101, 102, 108, 111–113, 126, 127, 129–131, 133, 135, 175, 179–187, 189–191, 194–198, 200, 205, 209–211, 215, 217, 221, 222, 229, 242, 264, 273, 375, 376, 378, 383, 386, 392–395, 398, 403–409, 413, 416, 417, 420, 421, 423, 431, 434, 436, 437, 443, 444, 447, 449, 450, 452, 471, 473–478, 480, 481, 484–488, 492, 493, 495, 496, 499, 500, 504, 507–509, 575–580, 596, 628, 633, 636, 642–645, 648, 653, 654, 656–659, 663, 664, 666–668, 671–674, 682, 685–687, 694, 695, 698, 700–702, 704, 723–730, 732, 733, 736–740, 744, 745, 747, 748, 751, 752, 756, 759, 761, 788, 828–833, 849, 882, 887, 890, 892, 897, 898, 900, 903, 909–915, 919, 920, 922–924, 927–930, 938, 942–945, 951, 952, 954, 955, 957, 958, 978–985, 987, 988, 991–993, 995, 997, 999, 1000, 1002–1004, 1006, 1007, 1012, 1018, 1030, 1033
\strut 279–282, 284, 286, 294, 295, 297, 305, 307, 309, 313, 314, 316, 323, 508, 581–584, 586, 588–590, 599–602, 606–608, 613–615, 617–619, 623, 760, 835–837, 839, 841–843, 852–856, 860–862, 867–869, 871–873, 877
style (option) 11
styles:
  altlist 280
  altlistgroup 280
  altlisthypergroup 280
  altlong4col 287
  altlong4col-booktabs 290
  altlong4colborder 287
  altlong4colheader 287
  altlong4colheaderborder 288
  altlongragged4col 296
  altlongragged4col-booktabs . 291
  altlongragged4colborder 298
  altlongragged4colheader 297

altlongragged4colheaderborder 298
altsuper4col 310
altsuper4colborder 311
altsuper4colheader 310
altsuper4colheaderborder . . . 311
altsuperragged4col 316
altsuperragged4colborder . . . 317
altsuperragged4colheader . . . 317
altsuperragged4colheaderborder 317
alttree 324
alttreegroup 327
alttreehypergroup 327
index 319
indexgroup 320
indexhypergroup 320
inline 275
list 278
listdotted 281
listgroup 279
listhypergroup 279
long 282
long-booktabs 288
long3col 283
long3col-booktabs 289
long3colborder 284
long3colheader 285
long3colheaderborder 285
long4col 285
long4col-booktabs 289
long4colborder 286
long4colheader 286
long4colheaderborder 286
longborder 283
longheader 283
longheaderborder 283
longragged 293
longragged-booktabs 290
longragged3col 295
longragged3col-booktabs 291
longragged3colborder 296
longragged3colheader 296
longragged3colheaderborder . 296
longraggedborder 294
longraggedheader 294
longraggedheaderborder 295
mcolalttree 303
mcolalttreegroup 303
mcolalttreehypergroup 303
mcolalttreespannav 304
mcolindex 299
mcolindexgroup 299
mcolindexhypergroup 299
mcolindexspannav 300
mcoltree 300
mcoltreegroup 301
mcoltreehypergroup 301
mcoltreenoname 301
mcoltreenonamegroup 302
mcoltreenonamehypergroup . . . 302
mcoltreenonamespannav 302
mcoltreespannav 301
sublistdotted 281
super 305
super3col 307
super3colborder 307
super3colheader 308
super3colheaderborder 308
super4col 308
super4colborder 309
super4colheader 309
super4colheaderborder 310
superborder 306
superheader 306
superheaderborder 306
superragged 312
superragged3col 314
superragged3colborder 315
superragged3colheader 315
superraggedborder 313
superraggedheader 313
superraggedheaderborder 314
superraggedright3colheaderborder 315
tree 321
treegroup 322
treehypergroup 322
treenoname 322
treenonamegroup 323
treenonamehypergroup 323
subentrycounter (option) 15
\subglossentry . . . 101, 216, 225, 234, 276, 279–282, 284, 286, 294, 295, 297, 305, 307, 309, 313, 314, 316, 320, 321, 323, 325, 431, 495, 501, 502, 507, 508, 598–602, 606–608, 613–615, 617–619, 621–624, 682,

747, 753, 759, 851–856, 860–862, 867–869, 871–873, 875, 876, 878, 938, 1003, 1009, 1014
\subitem 299, 319, 320, 585, 609, 620, 621, 838, 863, 874, 875
sublistdotted (style) 281
\subsubitem 299, 319, 320, 585, 609, 621, 838, 863, 874, 875
super (style) 305
super3col (style) 307
super3colborder (style) 307
super3colheader (style) 308
super3colheaderborder (style) . . . 308
super4col (style) 308
super4colborder (style) 309
super4colheader (style) 309
super4colheaderborder (style) . . . 310
superborder (style) 306
superheader (style) 306
superheaderborder (style) 306
superragged (style) 312
superragged3col (style) 314
superragged3colborder (style) . . . 315
superragged3colheader (style) . . . 315
superraggedborder (style) 313
superraggedheader (style) 313
superraggedheaderborder (style) . 314
superraggedright3colheaderborder (style) 315
symbol (key) 76
symbolaccess (key) 338
\symbolname . 44, 47, 286, 287, 290, 298, 309–311, 317, 318, 398, 399, 603, 604, 609, 615–617, 620, 648, 649, 856, 858, 862, 863, 869, 870, 873, 874, 903, 905
symbolplural (key) 77
symbolpluralaccess (key) 338
symbols (option) 40

T

\t 26, 387, 637, 891
\tablehead 305–318, 613–620, 867–874
\tabletail 305–318, 613–620, 867–874
\tabularnewline 282–287, 289, 290, 294–298, 305–318, 583, 584, 588, 589, 600–604, 606–609, 613–620, 837, 841, 842, 854–858, 860–863, 867–874
\texorpdfstring 167, 205, 469, 721, 973, 998
\text 4, 5, 879, 880
text (key) 76
textaccess (key) 338
\textbar 274, 597, 850, 1034
\textbf 240, 318, 507, 511, 584–587, 620, 759, 763, 838–840, 874, 1017
\textit 240, 511, 763, 1017
\textmd 240, 511, 763, 1017
\textrm 240, 511, 762, 1017
\textsc 240, 251, 252, 258, 371, 511, 517, 522, 526, 529, 531, 534, 568, 763, 768, 769, 774, 779, 782, 784, 787, 821, 1017, 1023, 1024, 1028
\textsf 240, 511, 763, 1017
\textsl 240, 511, 763, 1017
\textsmaller 252, 258, 372, 517, 522, 526, 530, 531, 534, 568, 769, 774, 779, 782, 784, 787, 821, 1023, 1024, 1029
\texttt 240, 511, 763, 1017
\textulc 242, 511, 763, 1018
\textup 240, 242, 511, 763, 1017, 1018
\TH 27, 388, 638, 892
\th 27, 388, 638, 892
\the 43, 46, 56, 61, 63, 72, 128–134, 181, 185, 187, 189, 201, 203, 207, 208, 212, 224, 239, 240, 247–251, 255, 257, 364, 368, 370, 373, 397, 399, 404, 407, 409, 413, 448–453, 474, 477, 479, 480, 488, 489, 491, 494, 501, 506, 507, 510, 513–516, 520, 521, 523–534, 536, 540, 561, 562, 565, 566, 569–573, 575, 576, 579, 648, 649, 654, 658, 659, 663, 699–705, 726, 729–732, 740, 741, 743, 746, 753, 758, 759, 762, 765–768, 771, 773, 775–779, 781, 783, 785–787, 789, 790, 793, 814, 815, 818, 819, 822–826, 828, 830, 832, 903, 904, 910, 913, 915, 919, 953–958, 980, 983, 985, 987, 995, 996, 998, 1001, 1009, 1017, 1020–1023, 1026, 1027
\the@numberlist 176, 472, 723, 724, 978
\theglossary 6, 376, 626, 880

theglossary (env.) 232
\theglossaryentry 14, 231, 381, 505, 630, 757, 885, 1013
\theglossarysubentry 15, 231, 381, 506, 631, 757, 885, 1013
\theglsentrycounter 125, 126, 205, 208, 446, 490, 491, 576, 697, 742, 743, 829, 950, 951, 998, 999
\theH 210, 492, 744, 1000
\theHglossaryentry . 14, 381, 630, 885
\theHglossarysubentry 15, 381, 631, 885
\theHglsentrycounter 125, 126, 205, 208, 446, 447, 490, 491, 697, 742, 743, 951, 998, 999
\thepage 135
\thesection 43, 397, 647, 903
\this@dialect 46, 328, 329, 399, 574, 591, 649, 827, 844, 905
\tiny 8, 378, 627, 881
\tl 5, 104, 135, 880
toc (option) 10
\toks@ 43, 46, 56, 61, 63, 72, 128–134, 187, 189, 212, 239, 240, 397–399, 404, 407, 409, 413, 448–453, 478–480, 494, 506, 507, 510, 540, 575, 648, 649, 654, 658, 659, 663, 699–705, 730–732, 746, 758, 759, 762, 793, 828, 903, 904, 910, 913, 915, 919, 953–958, 985, 987, 1001, 1016, 1017
\toprule 289, 290, 604, 857, 858
\TrackIfKnownLanguage 45, 904
\trans@languages . . 46, 398, 648, 904
\translate 46, 47, 399, 649, 650, 904, 905
translate (option) 30
\translatelet 19, 20, 40, 383, 384, 395, 396, 633, 634, 646, 887, 888, 901
tree (style) 321
treegroup (style) 322
treehypergroup (style) 322
treenoname (style) 322
treenonamegroup (style) 323
treenonamehypergroup (style) 323
\triangleright 8, 378, 627, 881
\ttfamily 8, 378, 627, 881
\TX@trial 105, 432, 683, 940
type (key) 77
\typeout 23, 385, 635, 889

U

\u 26, 387, 637, 891
\uccode 224, 501, 753, 1008
ucmark (option) 14
\undef 79, 85, 213, 416, 420, 494, 667, 671, 746, 923, 927, 1001
\unexpanded 41, 902
\unskip 94, 281, 426, 582, 600, 677, 835, 853, 933
\unvbox 292, 605, 606, 859
\usedictionary 46, 398, 649, 904
\usepackage 27, 221, 222, 242, 375, 376, 499, 500, 751, 752, 892, 1006, 1007, 1018
user1 (key) 79
user1access (key) 339
user2 (key) 79
user2access (key) 339
user3 (key) 79
user3access (key) 339
user4 (key) 79
user4access (key) 340
user5 (key) 79
user5access (key) 340
user6 (key) 80
user6access (key) 340
\UTFviii@two@octets 27, 388, 638, 892
\UTFviii@two@octets@combine 27, 388, 638, 892

V

\v 26, 388, 637, 892
\value 104, 939
\vbox 292, 605, 606, 859
\vsize 292, 606, 859
\vskip . . . 278, 292, 299, 318, 599, 605, 609, 620, 852, 859, 863, 874
\vss 292, 605, 606, 859

W

\warn@nomakeglossaries 193, 195, 196, 198, 484–486, 735–738, 990–993
\warn@noprintglossary 195–197, 213, 216, 484–486, 494, 495, 736–738, 746, 747, 991–993, 1001, 1003

\write 35, 86, 111, 181–186, 191–194, 197, 201, 217, 393, 420, 421, 436, 474–478, 482, 483, 486, 489, 496, 576–580, 643, 671, 686, 687, 726–730, 734, 735, 738, 740, 748, 830–833, 898, 927, 943, 944, 980–984, 988–991, 993, 996, 1003, 1004
writeglslabelnames (option) 36
writeglslabels (option) 36
\writeist 181, 190, 474, 477, 478, 481, 576, 579, 580, 726, 728, 730, 733, 830, 832, 833, 980, 983, 984, 988

X

\x 510
\xatlevel@ 125, 446, 697, 950
\xcapitalisewords 469
\xdef 98, 428, 430, 495, 596, 679, 849, 850, 935
\xglsaccsupp . . 346, 346, 545, 546, 799
\xglsfieldaccsupp 345, 545, 798
\xifinlist 84, 670, 926
\xifinlistcs 218, 219, 222, 497, 500, 748, 749, 752, 1004, 1005, 1007
xindy (option) 33
xindygloss (option) 34
xindynoglsnumbers (option) 34
\xmakefirstuc 437–439, 443, 467, 468, 592, 688–690, 694, 719, 720, 845
\xspace 241, 511, 763, 1017

Y

\year 181, 185, 474, 477, 576, 579, 726, 729, 830, 832, 980, 983

Z

\Z 235, 1014
\z@ 292, 605, 606, 859