

Documented Code For glossaries v4.51

Nicola L.C. Talbot

Dickimaw Books

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

2022-11-02

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	113
	2.11.1 Links to glossary entries	119
	2.11.2 Displaying entry details without adding information to the glossary	164
2.12	Adding an entry to the glossary without generating text	176
2.13	Creating associated files	177
2.14	Writing information to associated files	199
2.15	Glossary Entry Cross-References	209
2.16	Displaying the glossary	211
2.17	Acronyms	239
2.18	Predefined acronym styles	245
2.19	Predefined Glossary Styles	263
2.20	Debugging Commands	263
2.21	Compatibility with version 2.07 and below	269

3	Glossary Styles	269
3.1	Glossary hyper-navigation definitions (glossary-hypernav.sty) . . .	269
3.2	In-line Style (glossary-inline.sty)	272
3.3	List Style (glossary-list.sty)	275
3.4	Glossary Styles using longtable (glossary-long.sty)	279
3.5	Glossary Styles using longtable and booktabs (glossary-longbooktabs.sty)	285
3.6	Glossary Styles using longtable (glossary-longragged.sty)	290
3.7	Glossary Styles using multicols (glossary-mcols.sty)	296
3.8	Glossary Styles using supertabular environment (glossary-super.sty)	301
3.9	Glossary Styles using supertabular environment (glossary-superragged.sty)	308
3.10	Tree Styles (glossary-tree.sty)	315
4	Multi-Lingual Support	324
4.1	Babel Captions (glossaries-babel.sty)	324
4.2	Polyglossia Captions (glossaries-polyglossia.sty)	325
5	Prefix Support (glossaries-prefix Code)	326
6	Accessibility Support (glossaries-accsupp Code)	333
6.1	Defining Replacement Text	335
6.2	Accessing Replacement Text	339
6.3	Displaying the Glossary	359
6.4	Acronyms	361
6.5	Debugging Commands	371
7	Backwards Compatibility	372
7.1	glossaries-compatible-207	372
7.2	glossaries-compatible-307	372
8	Rollback	373
8.1	Rollback v4.46 (glossaries-2020-03-19.sty)	373
8.2	Rollback v4.46 (glossaries-accsupp-2020-03-19.sty)	536
8.3	Rollback v4.46 (glossaries-babel-2020-03-19.sty)	571
8.4	Rollback v4.46 (glossaries-compatible-207-2020-03-19.sty)	572
8.5	Rollback v4.46 (glossaries-compatible-307-2020-03-19.sty)	577
8.6	Rollback v4.46 (glossaries-polyglossia-2020-03-19.sty)	588
8.7	Rollback v4.46 (glossaries-prefix-2020-03-19.sty)	588
8.8	Rollback v4.46 (glossary-hypernav-2020-03-19.sty)	593
8.9	Rollback v4.46 (glossary-inline-2020-03-19.sty)	594
8.10	Rollback v4.46 (glossary-list-2020-03-19.sty)	595
8.11	Rollback v4.46 (glossary-list-2020-03-19.sty)	597
8.12	Rollback v4.46 (glossary-longbooktabs-2020-03-19.sty)	600
8.13	Rollback v4.46 (glossary-longragged-2020-03-19.sty)	603
8.14	Rollback v4.46 (glossary-mcols-2020-03-19.sty)	606
8.15	Rollback v4.46 (glossary-super-2020-03-19.sty)	610
8.16	Rollback v4.46 (glossary-superragged-2020-03-19.sty)	614

8.17 Rollback v4.46 (glossary-tree-2020-03-19.sty)	617
8.18 Rollback v4.49 (glossaries-2021-11-01.sty)	623
8.19 Rollback v4.49 (glossaries-accsupp-2021-11-01.sty)	789
8.20 Rollback v4.49 (glossaries-babel-2021-11-01.sty)	824
8.21 Rollback v4.49 (glossaries-compatible-207-2021-11-01.sty)	825
8.22 Rollback v4.49 (glossaries-compatible-307-2021-11-01.sty)	831
8.23 Rollback v4.49 (glossaries-polyglossia-2021-11-01.sty)	841
8.24 Rollback v4.49 (glossaries-prefix-2021-11-01.sty)	841
8.25 Rollback v4.49 (glossary-hypernav-2021-11-01.sty)	846
8.26 Rollback v4.49 (glossary-inline-2021-11-01.sty)	847
8.27 Rollback v4.49 (glossary-list-2021-11-01.sty)	849
8.28 Rollback v4.49 (glossary-long-2021-11-01.sty)	851
8.29 Rollback v4.49 (glossary-longbooktabs-2021-11-01.sty)	854
8.30 Rollback v4.49 (glossary-longragged-2021-11-01.sty)	857
8.31 Rollback v4.49 (glossary-mcols-2021-11-01.sty)	860
8.32 Rollback v4.49 (glossary-super-2021-11-01.sty)	863
8.33 Rollback v4.49 (glossary-superragged-2021-11-01.sty)	867
8.34 Rollback v4.49 (glossary-tree-2021-11-01.sty)	871

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` v4.51: L^AT_EX2e 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}
\DeclareCurrentRelease{v4.51}{2022-11-02}
```

Declare package:

```
\ProvidesPackage{glossaries}[2022/11/02 v4.51 (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 L^AT_EX3 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 `\glssupercase` instead.

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

Just treat blockers and mappings as exclusions.

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

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

}
\ExplSyntaxOff
```

```
\RequirePackage{xfor}
```

```
\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@glsgls@debug
\define@choicekey{glossaries.sty}{debug}{\glsgls@debug@val\glsgls@debug@nr}%
{true,false,showtargets,showaccsupp}[true]{%
\ifcase\glsgls@debug@nr\relax
% debug=true
\@glsgls@debugtrue
\renewcommand*{\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*{\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\let\@glsgls@showtarget\@gobble
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\or
% debug=false
\@glsgls@debugfalse
\let\@glsgls@showtarget\@gobble
\let\@glsgls@showaccsupp\@gobblethree
\PackageInfo{glossaries}{debug mode OFF}%
\or
% debug=showtargets
\@glsgls@debugtrue
\renewcommand*{\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*{\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsgls@showtarget}{\@glsgls@showtarget}%
\or
% debug=showaccsupp
\@glsgls@debugtrue
\renewcommand*{\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*{\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsgls@showaccsupp}{\glsgls@showaccsupp}%
\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]{%
\glsshowtargetsymbolsymbol\marginpar{\glsshowtargetsymbolsymbol\glsshowtargetfont #1}}

```

`\glsshowtargetsymbolsymbol`

```

\newcommand*{\glsshowtargetsymbolsymbol}{\tiny$\triangleright$}

```

`\glsshowtargetfont`

```

\newcommand*{\glsshowtargetfont}{\ttfamily\footnotesize}

```

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

```

\newcommand*{\@glsshowtarget}[1]{%

```

`\glsdoshowtarget` Make it easier to customise.

```

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

```

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

```

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

```


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

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

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

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

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

`\@gls@see@noindex`

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

`seenoinindex`

```
\define@choicekey{glossaries.sty}{seenoinindex}%
[\gls@seenoinindex@val\gls@seenoinindex@nr]{error,warn,ignore}{%
  \ifcase\gls@seenoinindex@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}{}%
  \else
    \PackageError{glossaries}{\@gls@see@noindex: invalid choice}%
  \fi}
```

```

\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 ignore its argument).

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

savenumberlist Provide means to store the number list for entries.

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

`\@glo@seeautonumberlist`

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

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

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

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

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

`\@gls@loadlong`

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

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

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

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

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

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

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

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

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

\@gls@loadlist

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

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

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

\@gls@loadtree

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

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

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

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

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

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

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

nopostdot Boolean option to suppress post description dot

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

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

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

ucmark Boolean option to determine whether or not to use upper case in definition of `\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}{%
  \ifglseentrycounter
    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]{}
\glseentrycounterfalse
```

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}%
  \glseentrycountertrue
  \@gls@define@glossaryentrycounter
}
```

`\@gls@counterwithin` The default value is no parent counter:
`\newcommand*{\@gls@counterwithin}{}`

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

`\define@glossarysubentrycounter`

```

\newcommand{\@gls@define@glossarysubentrycounter}{%
Check if counter already defined.
\ifundef\c@glossarysubentry
{%
\ifglssubentrycounter
\ifglsubentrycounter
\newcounter{glossarysubentry}[glossaryentry]%
\else
\newcounter{glossarysubentry}%
\fi
}
}

```

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

```

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

```

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

```

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

```

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

```

\newcommand{\glsindexingsetting}{\ifglxindy xindy\else makeindex\fi}

```

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

```

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

```

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

```

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

```

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

`\glsprestandardsort`

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

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

`\@glo@check@sortallowed`

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

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

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

Store entry information when it's defined.

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

No count register required for standard sort.

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

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

```
\def\@gls@defsort##1##2{%
  \ifx\@glo@sort\@glsdefaultsort
    \let\@glo@sort\@glo@name
  \fi

  \let\glsdosanitizesort\@gls@sanitizesort
  \glsprestandardsort{\@glo@sort}{##1}{##2}%
  \expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
}%
```

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

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

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

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

Set standard sort as the default:

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

`\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}%
    }
  \else
    \let\@gls@do@acronymsdef\relax
  \fi
}
```

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}{}
```

`\@addtoacronymlists`

```

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

```

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

```

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

```

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

`\glsIfListOfAcronyms`

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

```

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

```

Internal command requires label and list to be expanded:

```

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

```

Not found

```

    #4%
  \else

```

Found

```

    #3%
  \fi
}

```

`\if@glsisacronymlist` Convenient boolean.

```

\newif\if@glsisacronymlist

```

`\gls@checkisacronymlist` Sets the above boolean if argument is a label representing a list of acronyms.

```

\newcommand*{\gls@checkisacronymlist}[1]{%
  \glsIfListOfAcronyms{#1}%
  {\@glsisacronymlisttrue}{\@glsisacronymlistfalse}%
}

```

`\SetAcronymLists` Sets the “list of acronyms” list. Argument must be a comma-separated list of glossary labels. (Doesn’t check at this point if the glossaries exists.)

```

\newcommand*{\SetAcronymLists}[1]{%
  \renewcommand*{\@glsacronymlists}{#1}%
}

```

`acronymlists`

```

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

```

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

`\glscounter`

```

\newcommand{\glscounter}{page}

```

`counter` The counter option changes the default counter. (This just redefines `\glscounter`.)

```

\define@key{glossaries.sty}{counter}{%
  \renewcommand*{\glscounter}{#1}%
}

```

`\@gls@nohyperlist`

```

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

```

`\GlsDeclareNoHyperList`

```

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

```

nohypertypes

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

\GlossariesWarning Prints a warning message.

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

\GlossariesWarningNoLine Prints a warning message without the line number.

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

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

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

nowarn Define package option to suppress warnings

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

\@gls@missinglang@warn Missing language warning.

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

nolangwarn Suppress warning if language support not found.

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

```

\@gls@warnonglossdefined Issue a warning if overriding \printglossary
    \newcommand*{\@gls@warnonglossdefined}{%
        \GlossariesWarning{Overriding \string\printglossary}%
    }

\@gls@warnontheglossdefined Issue a warning if overriding theglossary
    \newcommand*{\@gls@warnontheglossdefined}{%
        \GlossariesWarning{Overriding ‘theglossary’ environment}%
    }

noredefwarn Suppress warning on redefinition of \printglossary
    \@gls@declareoption{noredefwarn}{%
        \renewcommand*{\@gls@warnonglossdefined}{}%
        \renewcommand*{\@gls@warnontheglossdefined}{}%
    }

```

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

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

`\glssetexpandfield`

`\glssetexpandfield{<field>}`

Sets field to always expand.

```

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

```

`\glssetnoexpandfield`

`\glssetnoexpandfield{<field>}`

Sets field to never expand.

```

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

```

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

`\glssetexpandfield{type}`

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

`\glssetnoexpandfield{desc}`


```

\gls@assign@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}

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

Sanitizing stuff:

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

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

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

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

```

\@@gls@noidx@nosanitizesort

```
\newcommand*{\@@gls@noidx@nosanitizesort}{%
  \ifdefined@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\@secondoftwo
  \let\@tabacckludge\@secondoftwo
  \expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
  \let'\@firstofone
  \let'\@firstofone
  \let~\@firstofone
  \let"\@firstofone
  \let\u\@firstofone
  \let\t\@firstofone
  \let\d\@firstofone
  \let\r\@firstofone
  \let=\@firstofone
  \let\.\@firstofone
  \let~\@firstofone
  \let\v\@firstofone
  \let\H\@firstofone
  \let\c\@firstofone
  \let\b\@firstofone

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

```

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

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

```

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

```

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

```

\@gls@removedoption

```

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

```

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

```

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

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

```

```

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

```

sanitizesort

```

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

```

Default setting:

```

\glsssanitizesorttrue
\glsssetnoexpandfield{sortvalue}%

```

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

```

\newcommand*{\@glS@noidx@setsanitizesort}{%
\glsssanitizesortfalse
\glsssetexpandfield{sortvalue}%
}
\define@choicekey[glS]{sanitize}{sort}{true,false}[true]{%
\setbool{glsssanitizesort}{#1}%
\ifglsssanitizesort
\glsssetnoexpandfield{sortvalue}%
\else
\glsssetexpandfield{sortvalue}%
\fi
\GlossariesWarning{sanitize={sort} package option

```

```

    deprecated. Use sanitizesort instead}%
}

```

sanitize

```

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

```

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

```

\newif\ifglstranslate

```

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

\@gls@usetranslator

```

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

```

\glsifusedtranslatordict Checks if given translator dictionary has been loaded.

```

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

```

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}}
```

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

```
\newif\ifglxindy
```


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@glsglnumberstrue
```

`\@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{\language}%
```

`\gls@xdy@set@lang` If the `nomain` option has been used, need to check `\glsgldefaulttype` 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@\glsgldefaulttype @language}
{\let\@gls@langmod\@xdy@main@language}%
{\letcs\@gls@langmod{@xdy@\glsgldefaulttype @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 ‘restoremakeloss’ has no effect
(\string\makeglossaries\space has already been used)}%
}%
{%
\PackageInfo{glossaries}{\string\makeglossaries\space and
\string\makenoidxglossaries\space have been restored}%
\let\@domakeglossaries\@firstofone
}%
}%
}

```

\do@write@glslabels

```

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

```

\@do@write@glslabels

```

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

```

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

```

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

```

\@write@glslabelsname

```

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

```

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

```

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

```

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

```

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

```

`\ifglautomake`

```

\newif\ifglautomake

```

`\glsl@automake@nr`

```

\newcommand{\glsl@automake@nr}{1}

```

`\glsl@automake@makegloss` Provide a way to allow `makeglossaries` or `makeglossaries-lite` run via `automake` instead of explicit calls to `makeindex` or `xindy`.

```

\newcommand{\glsl@automake@makegloss}{}

```

`\glsl@automake@option`

```

\newcommand{\glsl@automake@option}{automake=\glsl@automake@val}

```

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

```

\newcommand*{\@glsl@do@automake@err}{%
\PackageError{glossaries}{You must use
\string\makeglossaries\space with \glsl@automake@option}
{%
Either remove the \glsl@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}%
[\glsl@automake@val\glsl@automake@nr]{true,false,immediate,makegloss,lite,delayed}[immediate]{%
\ifcase\glsl@automake@nr\relax
\GlossariesWarning{automake=true deprecated. Use automake=delayed for a delayed write}%
case 0: true
\glslautomaketrue
\renewcommand{\glsl@automake@makegloss}{}%
\or
case 1: false
\glslautomakefalse
\renewcommand{\glsl@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
}

```



```

\fi
}

```

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

```

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

```

Default is not to define index glossary:

```

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

```

\@gls@indexdef \indexname isn't set by glossaries.

```

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

```

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

```

\define@choicekey{glossaries.sty}{mfirstuc}%
[\@gls@mfirstuc@val\@gls@mfirstuc@nr]%
{expanded,unexpanded}%
{%
\ifcase\@gls@mfirstuc@nr
% expanded
\renewcommand*{\glsmakefirstuc}[1]{\MFUsentencecase{#1}}%
\or
% unexpanded
\renewcommand*{\glsmakefirstuc}[1]{\MFUsentencecase{\unexpanded{#1}}}%
\fi
}

```

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

```

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

```

Now process options passed to the package:

```
\ProcessOptionsX
```

No longer load backward compatibility stuff as from v4.50. If LuaLaTeX and automake set, need to load shellesc. Note that if automake is passed to `\setupglossaries` instead, then shellesc will need to be loaded explicitly.

```
\ifglautomake
\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,nottranslate,languages,nolong,nosuper,notree,nostyles,%
nomain,noglossaryindex}
```

Now define `\setupglossaries`:

```
\newcommand*{\setupglossaries}[1]{%
\renewcommand*{\@gls@setacrstyle}{}%
\ifglacrshortcuts
\def\@gls@setupshortcuts{\glsacrshortcuts}%
\else
\def\@gls@setupshortcuts{%
\ifglacrshortcuts
\DefineAcronymSynonyms
\fi
}%
\fi
\glsacrshortcutsfalse
\let\@gls@do@numbersdef\relax
\let\@gls@do@symbolssdef\relax
\let\@gls@do@indexdef\relax
\let\@gls@do@acronymsdef\relax
\ifglentrycounter
\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}

```

```

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

```

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

```

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

```

```

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

```

```

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

```

```

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

```

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

```

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

```

```

\glnumbersgroupname
\providecommand*{\glnumbersgroupname}{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-#1.1df}{\input{glossaries-#1.1df}}{}%
}
```

`\ProvidesGlossariesLang`

```
\newcommand*{\ProvidesGlossariesLang}[1]{%
  \ProvidesFile{glossaries-#1.1df}%
}
```

`\addglossarytocaptions` Does nothing if translator hasn't been loaded.

```
\newcommand*{\addglossarytocaptions}[1]{}
```

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

`\ifglstranslate`

Load tracklang:

```
\RequirePackage{tracklang}[2019/11/30]
```

Track any languages explicitly requested via the languages option.

```
\ifdefvoid\@gls@requested@languages{%
  {%
    \for\@gls@thislang:=\@gls@requested@languages\do{%
      \TrackIfKnownLanguage{\@gls@thislang}{}%
      {\GlossariesWarning{unknown language '\@gls@thislang'}}%
    }%
  }%
}
```

Load translator if required.

```
\@gls@usetranslator
```

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

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

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

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

Check for tracked languages

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

```

}%
{}%
if using translator use translator interface.
\glsifusetranslator
{%
  \renewcommand*{\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*{\glossaryname}{\translate{Glossary}}%
  \renewcommand*{\acronymname}{\translate{Acronyms}}%
  \renewcommand*{\entryname}{\translate{Notation (glossaries)}}%
  \renewcommand*{\descriptionname}{%
    \translate{Description (glossaries)}}%
  \renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}%
  \renewcommand*{\pagelistname}{%
    \translate{Page List (glossaries)}}%
  \renewcommand*{\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.

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

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

```
\@onlypremakeg\setStyleFile
```

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

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

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

The `makeglossaries` Perl script picks up this name from the auxiliary file. If the name ends with `.xdy` it calls `xindy` otherwise it calls `makeindex`. Since its not required by L^AT_EX, `\@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 $\langle letter \rangle \langle compositor \rangle \langle number \rangle$. 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.

```
\ifglxindy
  \newcommand*\glsSetAlphaCompositor[1]{%
    \renewcommand*\@glsAlphacompositor{#1}}
\else
  \newcommand*\glsSetAlphaCompositor[1]{%
    \glsnxindywarning\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]{\glsnumber{#1}}%
}
```

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

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

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

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

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

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

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

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

```
\setglossarypreamble[<type>]{<text>}
```

`\setglossarypreamble`

Code provided by Michael Pock.

```
\newcommand{\setglossarypreamble}[2][\glsdefaultttype]{%
  \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{\gls glossarymark}[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.

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

`\@gls@ifinlist`

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

`\GlsAddXdyCounters` Need to know all the counters that will be used in location numbers for Xindy. Argument may be a single counter name or a comma-separated list of counter names.

```
\ifglxindy
\newcommand*\@xdycounters{}\@glscounter}
\newcommand*\GlsAddXdyCounters[1]{%
\@for\@gls@ctr:=#1\do{%
```

Check if already in list before adding.

```
\edef\@do@addcounter{%
\noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
{%
\noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
\noexpand\@gls@ctr}%
}%
}%
\@do@addcounter
}
```

Only has an effect before `\writeist`:

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

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

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

```

occurrences of \string\GlsAddXdyCounters\space before the first
instance of \string\GlsAddXdyAttribute}%
}

```

`\GlsAddXdyAttribute` Adds an attribute.

```
\ifglxsindy
```

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

```
\newcommand*\@glssaddxdyattribute[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\@glssaddxdyattribute{#1}{\@this@counter}%
}
\gls@do@addxdyattribute
}%

```


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

```
\let\GlsAddXdyCounters\@disabled@glssaddxdycounters
}
```

Only has an effect before `\writeist`:

```
\@onlypremake\GlsAddXdyAttribute
\else
\newcommand*\GlsAddXdyAttribute[1]{%
\glsnxindywarning\GlsAddXdyAttribute}
\fi
```

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

```
\ifglxindy
\newcommand*{\@glss@addpredefinedattributes}{%
\GlsAddXdyAttribute{glssnumberformat}
\GlsAddXdyAttribute{textrm}
\GlsAddXdyAttribute{textsf}
\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glshypernumber}
\GlsAddXdyAttribute{hyperarm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hypertt}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}
\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}

\GlsAddXdyAttribute{glssignore}
}
\else
\let\@glss@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.

```
\ifglxindy
\newcommand*\GlsAddXdyAlphabet[2]{%
```

```

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

```

This code is only required for xindy:

```
\ifglsxindy
```

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

```

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

```

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

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

```

\protected@edef\@gls@roman{\@roman{0}\string"
  \string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@gls@roman
\edef\@gls@tmp{\string" \string"roman-numbers-lowercase\string"
  :sep \string"}%
\@onelevel@sanitize\@gls@tmp
\ifx\@gls@tmp\@gls@roman
  \expandafter
  \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
    \string"roman-numbers-lowercase\string"%
  }%
\else
  \expandafter
  \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{
    :sep \string"\@gls@roman\string"%
  }%
\fi

```

`\@Lclass@Roman-page-numbers` Upper case Roman numerals (I, II, ...).

```

\expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
  \string"roman-numbers-uppercase\string"%
}%

```

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

```

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

```

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

```

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

```

`\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 \@glsAlphacompositor\string"
  \string"arabic-numbers\string"%
}

```

`\Lclass@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 \@glscompositor\string"
  \string"arabic-numbers\string"%
}%

```

`\@xdyuserlocationdefs` List of additional location definitions (separated by `^^J`)

```

\def\@xdyuserlocationdefs{}

```

`\@xdyuserlocationnames` List of additional user location names

```

\def\@xdyuserlocationnames{}

```

End of xindy-only block:

```

\fi

```

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

```

\ifglxindy
  \newcommand\@xdycrossrefhook{}
\fi

```

`\GlsAddXdyLocation` `\GlsAddXdyLocation[<prefix-loc>]{<name>}{<definition>}` Define a new location called *<name>*. The definition must use xindy syntax. (Note that this

doesn't check to see if the location is already defined. That is left to xindy to complain about.)

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

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

Only has an effect before \writeist:

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

\@xdylocationclassorder Define location class order

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

Change the location order.

`\GlsSetXdyLocationClassOrder`

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

`\@xdysortrules` Define sort rules

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

`\GlsAddSortRule` Add a sort rule

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

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

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

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

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

`\GlsSetXdyStyles` Reset the list of required styles

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

```

```

\glsnoxywarning\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][]{%
\glsnoxywarning\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.

```

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

```

`\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 $\langle type \rangle$ is the glossary label and $\langle cmd \rangle$ is a control sequence which will be set to the entry label in the current iteration.

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

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

```
\foralllgsentries[glossary list]{ $\langle cmd \rangle$ }{ $\langle code \rangle$ }
```

Within **\foralllgsentries**, the current glossary type is given by **\@this@glo@**.

```
\newcommand*\foralllgsentries[3][\@glo@types]{%
  \expandafter\foralllglossaries\expandafter[#1]{\@this@glo@}%
  {%
    \forlgsentries[\@this@glo@]{#2}{#3}%
  }%
}
```

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

```
\ifglossaryexists{ $\langle type \rangle$ }{ $\langle true-text \rangle$ }{ $\langle false-text \rangle$ }
```

where $\langle type \rangle$ is the glossary's label. The unstarred form will do $\langle false-text \rangle$ for ignored glossaries. The starred form will do $\langle true-text \rangle$ for ignored glossaries.

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

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

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

\s@ifglossaryexists Starred form includes ignored glossaries.

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

Since the label is used to form the name of control sequences, by default UTF8 etc characters can't be used in the label. A possible workaround is to use **\scantokens**, but commands such as **\glentrytext** will no longer be usable in sectioning, caption etc commands. If the user really wants to be able to

construct a label with UTF8 characters, allow them the means to do so (but on their own head be it, if they then use entries in `\section` etc). This can be done via:

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

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

`\glsdetoklabel`

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

`\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 exist, otherwise do `<code>`.

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

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

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

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

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

`\glsdoifexistsorwarn`

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

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

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

`\glsdoifexistsordo`

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

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

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

`\doifglossarynoexistsordo`

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

`\ifglshaschildren`

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

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

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

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

`\ifglshasparent`

```
\newcommand{\ifglshasparent}[3]{%
  \glsdoifexists{#1}%
  {%
    \ifcsempy{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
  }%
}
```

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

```
\newcommand*{\ifglshasdesc}[3]{%
  \ifcsempy{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@glstetoklabel{#1}@desc}{@no@post@desc}%
  {#2}%
  {#3}%
}
```

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

```
\newrobustcmd*{\ifglshassymbol}[3]{%
  \letcs{\@glo@symbol}{glo@glstetoklabel{#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@glstetoklabel{#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@glstetoklabel{#1}@short}%
  \ifdefempty\@glo@short
  {#3}%
  {%
    \ifdefequal\@glo@short\@gls@default@value
    {#3}%
    {#2}%
  }%
}
```

`\ifgl$fieldvoid{<field>}{<label>}{<true part>}{<false part>}`

`\ifgl$fieldvoid`

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

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

```
\ifglshasfield{<field>}{<label>}{<true part>}{<false  
part>}
```

\ifglshasfield

```
\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

```
\ifdequal\@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@glstetoklabel{#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@forbidtextext\csname @glotype@#2@log\endcsname
\expandafter\@gls@forbidtextext\csname @glotype@#2@in\endcsname
\expandafter\@gls@forbidtextext\csname @glotype@#2@out\endcsname

```

Store the title:

```

\expandafter\def\csname @glotype@#2@title\endcsname{#5}%

\@gls@provide@newglossary
\protected@write\@auxout{}\string\@newglossary{#2}{#1}{#3}{#4}}%

```

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

```

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

```

Define sort counter if required:

```

\@gls@defsortcount{#2}%

```

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

```

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

```

`\altnewglossary`

```

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

```

Only define new glossaries in the preamble:

```

\@onlypreamble{\newglossary}

```

Only define new glossaries before `\makeglossaries`

```

\@onlypremakeg\newglossary

```

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

`\@newglossary`

```

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

```

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

`\@gls@setcounter`

```
\def\@gls@setcounter#1[#2]{%
  \expandafter\def\csname @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 $\langle name \rangle \langle description \rangle$.

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

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

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

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

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

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

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

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

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

`\@gls@default@value`

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

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

`\gls@set@xr@key`

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{\@gls@nonextpages}%
    \@gls@savenonumberlist{true}%
  \else
    \def\@glo@prefix{\@gls@nextpages}%
    \@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
}%

\@gls@noidx@prenumberlist

\newcommand{\@gls@noidx@prenumberlist}[1]{%
  \csuse\@glo@\@gls@detoklabel{#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}}
```

`\@glsnodelsc` Define command to generate error if description key is missing.

```
\newcommand*{\@glsnodelsc{%  
  \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
}
```

`\@gls@defaultsort` Define command to set default sort.

```
\newcommand*\@gls@defaultsort*\@glo@name}
```

`\gls@level` Register to increment entry levels.

```
\newcount\gls@level
```

`\@gls@noexpand@field`

```
\newcommand*\@gls@noexpand@field}[3]{%
  \expandafter\global\expandafter
  \let\csname glo@#1@#2\endcsname#3%
}
```

`\@gls@noexpand@fields`

```
\newcommand*\@gls@noexpand@fields}[4]{%
  \ifcsdef{gls@assign@#3@field}
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
    }%
    {%
      \csuse{gls@assign@#3@field}{#2}{#4}%
    }%
  }%
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%

      \@gls@noexpand@field{#2}{#3}{\@gls@value}%
    }%
    {%
```

```

        \@@gls@noexpand@field{#2}{#3}{#4}%
      }%
    }%
  }

\@@gls@expand@field
\newcommand{\@@gls@expand@field}[3]{%
  \expandafter
  \protected@xdef\csname glo@#1@#2\endcsname{#3}%
}

@gls@expand@fields
\newcommand{\@gls@expand@fields}[4]{%
  \ifcsdef{gls@assign@#3@field}
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
    }%
    {%
      \expandafter\@gls@startswithexpandonce#4\relax\relax@gls@endcheck
      {%
        \@@gls@expand@field{#2}{#3}{#4}%
      }%
      {%
        \csuse{gls@assign@#3@field}{#2}{#4}%
      }%
    }%
  }%
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \@@gls@expand@field{#2}{#3}{#1}%
    }%
    {%
      \@@gls@expand@field{#2}{#3}{#4}%
    }%
  }%
}

\@gls@startswithexpandonce
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2@gls@endcheck#3#4{%
  \def\@gls@tmp{#1}%
  \ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}

```

```
\gls@assign@field{<def value>}{<label>}{<field>}{<tmp cs>}
```

`\gls@assign@field`

Assigns an entry field. Expansion performed by default (except for name, symbol and description where backward compatibility required). If `<tmp cs>` is `<@gls@default@value>`, `<def value>` is used instead.

```
\let\gls@assign@field\@gls@expand@fields
```

`\glsexpandfields` Fully expand values when assigning fields (except for specific fields that are overridden by `\glssetnoexpandfield`).

```
\newcommand*{\glsexpandfields}{%
  \let\gls@assign@field\@gls@expand@fields
}
```

`\glsnoexpandfields` Don't expand values when assigning fields (except for specific fields that are overridden by `\glssetexpandfield`).

```
\newcommand*{\glsnoexpandfields}{%
  \let\gls@assign@field\@gls@noexpand@fields
}
```

`\newglossaryentry` Define `\newglossaryentry {<label>} {<key-val list>}`. There are two required fields in `<key-val list>`: name (or parent) and description. (See above.)

```
\newrobustcmd{\newglossaryentry}[2]{%
```

Check to see if this glossary entry has already been defined:

```
\glsdoifnoexists{#1}%
{%
  \gls@defglossaryentry{#1}{#2}%
}%
}
```

`\gls@defdocnewglossaryentry` The definition of `\newglossaryentry` is changed at the start of the document environment. As from v4.47 the see key now works in the document environment.

```
\newcommand*{\gls@defdocnewglossaryentry}{%
  \let\newglossaryentry\new@glossaryentry
}
```

`\provideglossaryentry` Like `\newglossaryentry` but does nothing if the entry has already been defined.

```
\newrobustcmd{\provideglossaryentry}[2]{%
  \ifglentryexists{#1}%
  {}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\@onlypreamble{\provideglossaryentry}
```

`\gls@glsdefs@labellist` List of labels of entries defined in `.glsdefs` file.

```
\newcommand*\gls@glsdefs@labellist{}\}
```

`\gls@glsdefs@addlabel` Save label of entry defined in `.glsdefs` file.

```
\newcommand*\gls@glsdefs@addlabel}[1]{\listxadd{\gls@glsdefs@labellist}{#1}}
```

`\gls@docdef@labellist` List of labels of entries defined in the document environment.

```
\newcommand*\gls@docdef@labellist{}\}
```

`\gls@docdef@addlabel` Save label of entry defined in the document environment.

```
\newcommand*\gls@docdef@addlabel}[1]{\listxadd{\gls@docdef@labellist}{#1}}
```

`\new@glossaryentry` For use in document environment. This opens the `.glsdefs` file, if not already open, so that the entry definition can be saved for the next \LaTeX run. This means that any glossaries at the start of the document can access the entry information.

```
\newrobustcmd{\new@glossaryentry}[2]{%
  \ifundef\@gls@deffile
  {%
    \global\newwrite\@gls@deffile
    \immediate\openout\@gls@deffile=\jobname.glsdefs
  }%
  {}%
```

Don't check for existence since if the entry has been defined in the `.glsdefs` it will need to be redefined in case the entry has been updated, but keep track of which entries have been defined in the document environment to prevent duplicate definitions.

```
\ifglentryexists{#1}%
{%
  \xifinlist{#1}{\gls@docdef@labellist}%
  {%
    This entry has already been defined in the document environment.
    \GlossariesWarning{Entry ‘#1’ has already been defined in the
      document environment (ignoring new definition)}%
    }%
    {%
      \xifinlist{#1}{\gls@glsdefs@labellist}%
      {%
        This entry was defined in the .glsdefs file so update the definition in case it
        has been changed.
        \gls@defglossaryentry{#1}{#2}%
        \gls@docdef@addlabel{#1}%
        \@gls@writedef{#1}%
        }%
        {%
```

This entry wasn't defined in the `.glsdefs` file so it must have been defined in the preamble.

```

\GlossariesWarning{Entry '#1' has already been defined in
the preamble (in which case remove the duplicate definition)
or you have just updated to glossaries.sty
v4.47 (in which case delete '\jobname.glsdefs')}}%
}%
}%
}%
{%

```

This entry hasn't been defined (new entry since last build).

```

\gls@defglossaryentry{#1}{#2}%
\gls@docdef@addlabel{#1}%
\@gls@writedef{#1}%
}%
}

```

At the start of the document input the `.glsdefs` file if it exists. This is now done by `\gls@begindocdefs`, which is redefined by `glossaries-extra`, so that this step can be skipped to avoid loading an obsolete `.glsdefs` file if the user switches to `glossaries-extra` with `docdef=restricted`.

```

\AtBeginDocument{\gls@begindocdefs}

```

The end of the document needs to check if the `.glsdefs` file has been opened, in which case it needs to be closed.

```

\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}

```

`\gls@begindocdefs` Input the `.glsdefs` file if it exists and enable document definitions if permitted.

```

\newcommand*{\gls@begindocdefs}{%
\@gls@enablesavenonumberlist
\edef\@gls@restoreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
\makeatletter
\InputIfFileExists{\jobname.glsdefs}{-}{-}%
\@gls@restoreat
\undef\@gls@restoreat
\gls@defdocnewglossaryentry
}

```

`\glsdefs@newdocentry` Used to define entries in the `.glsdefs` file.

```

\newcommand*{\glsdefs@newdocentry}[2]{%
\ifglentryexists{#1}%
{\GlossariesWarning{entry '#1' has already been defined in the preamble
(if you have moved the definition there from the document
environment, delete \jobname.glsdefs)}}%
{%
\gls@glsdefs@addlabel{#1}%
\gls@defglossaryentry{#1}{#2}%
}%
}

```

`\@gls@writedef` Writes glossary entry definition to `\@gls@deffile`. Allow for the possibility that the entry definition may have been moved from the document environment to the preamble, so warn rather than generate an error.

```
\newcommand*{\@gls@writedef}[1]{%
  \immediate\write\@gls@deffile
  {%
    \string\glsdefs@newdocentry{\glsdetoklabel{#1}}\glspercentchar^^J%
    \expandafter\@gobble\string\{\glspercentchar%
  }%
```

Write key value information:

```
\@for\@gls@map:=\@gls@keymap\do
{%
  \letcs\glo@value{glo@\glsdetoklabel{#1}}\expandafter\@secondoftwo\@gls@map}%
  \ifdef\glo@value
  {%
    \@onelevel@sanitize\glo@value
    \immediate\write\@gls@deffile
    {%
      \expandafter\@firstoftwo\@gls@map
      =\expandafter\@gobble\string\{\glo@value\expandafter\@gobble\string\},%
      \glspercentchar
    }%
  }%
  {}%
```

Provide hook:

```
\gls.writedefhook
\immediate\write\@gls@deffile
{%
  \glspercentchar^^J%
  \expandafter\@gobble\string\}\glspercentchar^^J%
}%
}
```

`\@gls@keymap` List of entry definition key names and corresponding tag in control sequence used to store the value.

```
\newcommand*{\@gls@keymap}{%
  {name}{name},%
  {sort}{sortvalue},% unescaped sort value
  {type}{type},%
  {first}{first},%
  {firstplural}{firstpl},%
  {text}{text},%
  {plural}{plural},%
  {description}{desc},%
  {descriptionplural}{descplural},%
  {symbol}{symbol},%
  {symbolplural}{symbolplural},%
```

```

{user1}{useri},%
{user2}{userii},%
{user3}{useriii},%
{user4}{useriv},%
{user5}{userv},%
{user6}{uservi},%
{long}{long},%
{longplural}{longpl},%
{short}{short},%
{shortplural}{shortpl},%
{counter}{counter},%
{parent}{parent}%
}

```

```
\@gls@fetchfield{<cs>}{<field>}
```

`\@gls@fetchfield`

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{<key>}{<default value>}{<no link cs>}
```

`\glsaddstoragekey`

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}%
}%
}%
}%
\@gls@saddstoragekey{#1}%
}

```

Unstarred version doesn't override default expansion.

```
\newcommand*{\@gls@saddstoragekey}[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}{}%
}%
}

```

```

\gls@saddkey{<key>}{<default value>}{<no link cs>}{<no link
ucfirst
cs>}{<link cs>}{<link ucfirst cs>}{<link allcaps cs>}

```

\gls@saddkey

Allow user to add their own custom keys.

```
\newcommand*{\gls@saddkey}{\ifstar\@sgls@saddkey\@gls@saddkey}
```

Starred version switches on expansion for this key.

```

\newcommand*{\@sgls@saddkey}[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}{}%
}%
}
```

`\glsfielddedef`

```
\glsfielddedef{<label>}{<field>}{<definition>}
```

```
\newrobustcmd{\glsfielddedef}[3]{%
\glsdoifexists{#1}%
{%
\protected@edef\@glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\@glo@label @#2}%
{%
\protected@csedef{glo@\@glo@label @#2}{#3}%
}%
}%
\PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
}%
}
```

`\glsfieldgdef`

```
\glsfieldgdef{<label>}{<field>}{<definition>}
```

```
\newrobustcmd{\glsfieldgdef}[3]{%
\glsdoifexists{#1}%
{%
\protected@edef\@glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\@glo@label @#2}%
}
```

```

    {%
      \expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
    }%
    {%
      \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
    }%
  }%
}

```

\glsfielddef

`\glsfielddef{<label>}{<field>}{<definition>}`

```

\newrobustcmd{\glsfielddef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
    }%
    {%
      \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
    }%
  }%
}

```

\glsfieldfetch

`\glsfieldfetch{<label>}{<field>}{<cs>}`

Fetches the value of the given field and stores in the given control sequence.

```

\newrobustcmd{\glsfieldfetch}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \letcs#3{glo@\@glo@label @#2}%
    }%
    {%
      \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
    }%
  }%
}

```

\glsunexpandedfieldvalue

`\glsunexpandedfieldvalue{<label>}{<field>}`

For use in expandable contexts where the field value is required, but the contents should not be expanded. The $\langle field \rangle$ 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{<label>}{<field>}{<string>}{<true>}{<false>}
```

\backslash ifglsfieldeq

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{<label>}{<field>}{<command>}{<true>}{<false>}
```

\backslash ifglsfielddefeq

Tests if the value of the given field is equal to the replacement text of the given command.

```
\newrobustcmd{\ifglsfielddefeq}[5]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
```

Use \backslash ifdefstrequal not \backslash ifcsstrequal as the #3 argument should be a command.

```
\expandafter\ifdefstrequal\csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}
```

```
\ifglsfieldcseq{<label>}{<field>}{<cs name>}{<true>}
{<false>}
```

`\ifglsfieldcseq`

As above but uses `\ifcsstrequal` instead of `\ifdefstrequal`

```
\newcommand{\ifglsfieldcseq}[5]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\glo@label @#2}%
    {%
      \ifcsstrequal{glo@\glo@label @#2}{#3}{#4}{#5}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
```

`\gls.writedefhook`

```
\newcommand*{\gls.writedefhook}{}%
```

`\gls@assign@desc`

```
\newcommand*{\gls@assign@desc}[1]{%
  \gls@assign@field{#1}{desc}{\@glo@desc}%
  \gls@assign@field{\@glo@desc}{#1}{descplural}{\@glo@descplural}%
}
```

`\longnewglossaryentry`

```
\newcommand{\longnewglossaryentry}[3]{%
  \glsdoifnoexists{#1}%
  {%
    \bgroup
    \let\@org@newglossaryentryprehook\@newglossaryentryprehook
    \long\def\@newglossaryentryprehook{%
      \long\def\@glo@desc{#3\leavevmode\unskip\nopostdesc}%
      \@org@newglossaryentryprehook
    }%
    \renewcommand*{\gls@assign@desc}[1]{%
      \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
      \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@desc}%
    }
    \gls@defglossaryentry{#1}{#2}%
  \egroup
}
```

Only allowed in the preamble. (Otherwise a long description could cause problems when writing the entry definition to the temporary file.)

```
\@onlypreamble{\longnewglossaryentry}
```

`\longprovideglossaryentry` As the above but only defines the entry if it doesn't already exist.

```
\newcommand{\longprovideglossaryentry}[3]{%
  \ifglseentryexists{#1}{}%
  {\longnewglossaryentry{#1}{#2}{#3}}%
}
\@onlypreamble{\longprovideglossaryentry}
```

`\gls@defglossaryentry{<label>}{<key-val list>}`

`\gls@defglossaryentry`

Defines a new entry without checking if it already exists.

```
\newcommand{\gls@defglossaryentry}[2]{%
```

Prevent any further use of `\GlsSetQuote`:

```
\let\GlsSetQuote\gls@nosetquote
```

Store label

```
\protected@edef\@glo@label{\glsdetoklabel{#1}}%
```

Provide a means for user defined keys to reference the label:

```
\let\glslabel\@glo@label
```

Set up defaults. If the name or description keys are omitted, an error will be generated.

```
\let\@glo@name\@gls@noname
```

```
\let\@glo@desc\@gls@nodesc
```

```
\let\@glo@descplural\@gls@default@value
```

```
\let\@glo@type\@gls@default@value
```

```
\let\@glo@symbol\@gls@default@value
```

```
\let\@glo@symbolplural\@gls@default@value
```

```
\let\@glo@text\@gls@default@value
```

```
\let\@glo@plural\@gls@default@value
```

Using `\let` instead of `\def` to make later comparison avoid expansion issues.
(Thanks to Ulrich Diez for suggesting this.)

```
\let\@glo@first\@gls@default@value
```

```
\let\@glo@firstplural\@gls@default@value
```

Set the default sort:

```
\let\@glo@sort\@gls@default@value
```

Set the default counter:

```
\let\@glo@counter\@gls@default@value
```

```
\def\@glo@see{}%
```

```
\def\@glo@parent{}%
```

```
\def\@glo@prefix{}
```

Initialise nonnumberlist setting if we're in the document environment.

```
\@gls@initnonnumberlist
```

```
\def\@glo@useri{}%
```

```
\def\@glo@userii{}%
```

```
\def\@glo@useriii{}%
```

```
\def\@glo@useriv{}%
```

```
\def\@glo@userv{}%
```

```
\def\@glo@uservi{}%
```

```
\def\@glo@short{}%
```

```
\def\@glo@shortpl{}%
```

```
\def\@glo@long{}%
```

```
\def\@glo@longpl{}%
```

Add start hook in case another package wants to add extra keys.

```
\@newglossaryentryprehook
```

Extract key-val information from third parameter:

```
\setkeys{glossentry}{#2}%
```

Check there is a default glossary.

```
\ifundef\glsdefaulttype
```

```
{%
```

```
\PackageError{glossaries}%
```

```
{No default glossary type (have you used 'nomain' by mistake?)}%
```

```
{If you use package option 'nomain' you must define
```

```
a new glossary before you can define entries}%
```

```
}%
```

```
{}%
```

Assign type. This must be fully expandable

```
\gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
```

```
\protected@edef\@glo@type{\glsentrytype{\@glo@label}}%
```

Check to see if this glossary type has been defined, if it has, add this label to the relevant list, otherwise generate an error.

```
\ifcsundef{glolist@\@glo@type}%
```

```
{%
```

```
\PackageError{glossaries}%
```

```
{Glossary type '\@glo@type' has not been defined}%
```

```
{You need to define a new glossary type, before making entries
```

```
in it}%
```

```
}%
```

```
{%
```

Check if it's an ignored glossary

```
\ifignoredglossary\@glo@type
```

```
{%
```


The description may be omitted for an entry in an ignored glossary.

```

\ifx\@glo@desc\@glsnodel
\let\@glo@desc\@empty
\fi
}%
{%
}%
\protected@edef\@glo@list{\csname glo@list\@glo@type\endcsname}%

\expandafter\protected@xdef\csname glo@list\@glo@type\endcsname{%
\@glo@list{\@glo@label},}%
}%

```

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{\@glo@type\@glo@type @counter}%
}%
\gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
\gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
\gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
\gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
\gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
\gls@assign@field{}{\@glo@label}{userv}{\@glo@userv}%
\gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
\gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
\gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
\gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
\gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%
\ifx\@glo@name\@glsnoname
  \@glsnoname
  \let\@glo@name\@gls@default@value
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%

```

Set default numberlist if not defined:

```

\ifcsundef{glo@\@glo@label @numberlist}%
{%
  \csxdef{glo@\@glo@label @numberlist}{%
    \noexpand\@gls@missingnumberlist{\@glo@label}}%
}%
{}%

```

Store nonumberlist setting if we're in the document environment.

```

\@gls@storenonumberlist{\@glo@label}%

```

The smaller and smallcaps options set the description to \@glo@first. Need to check for this, otherwise it won't get expanded if the description gets sanitized.

```

\def\@glo@@desc{\@glo@first}%
\ifx\@glo@desc\@glo@@desc
  \let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@glsnodesc
  \@glsnodesc
  \let\@glo@desc\@gls@default@value
\fi
\gls@assign@desc{\@glo@label}%

```

Set the sort key for this entry:

```

\@gls@defsort{\@glo@type}{\@glo@label}%

\def\@glo@@symbol{\@glo@text}%
\ifx\@glo@symbol\@glo@@symbol
  \let\@glo@symbol\@glo@text
\fi

```

```

\gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
\expandafter
\gls@assign@field\expandafter
{\csname glo@\@glo@label @symbol\endcsname}
{\@glo@label}{symbolplural}{\@glo@symbolplural}%

```

Define an associated boolean variable to determine whether this entry has been used yet (needs to be defined globally):

```

\expandafter\protected@xdef\csname glo@\@glo@label @flagfalse\endcsname{%
\noexpand\global
\noexpand\let\expandafter\noexpand
\csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagtrue\endcsname{%
\noexpand\global
\noexpand\let\expandafter\noexpand
\csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
}%
\csname glo@\@glo@label @flagfalse\endcsname

```

Sort out any cross-referencing if required.

```

\@glo@autosee

```

Determine and store main part of the entry's index format.

```

\ifignoredglossary\@glo@type
{%
\csdef{glo@\@glo@label @index}{}%
}
{%
\do@glo@storeentry{\@glo@label}%
}%

```

Define entry counters if enabled:

```

\@newglossaryentry@defcounters

```

Add end hook in case another package wants to add extra keys.

```

\@newglossaryentryposthook
}

```

`\@glo@autosee` Automatically implement `\glssee`.

```

\newcommand*{\@glo@autosee}{%
\ifdefined\@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{,}%
  \forlslentries[\glo@type]{\glo@label}%
  {%
    \ifdefequal\@glo@thislabel\glo@label
      {\protected@eappto\glo@list{\glo@label,}}%
    }%
  \cslet\glo@list@\glo@type{\glo@list}%
  \csdef{glo@\@glo@thislabel @type}{#2}%
}

\@glossaryentryfield Indicate what command should be used to display each entry in the glossary.
(This enables the glossaries-accsupp package to use \accsuppglossaryentryfield
instead.)
\ifglxindy
  \newcommand*{\@glossaryentryfield}{\string\glossentry}
\else
  \newcommand*{\@glossaryentryfield}{\string\glossentry}
\fi

\@glossarysubentryfield Indicate what command should be used to display each subentry in the glossary.
(This enables the glossaries-accsupp package to use \accsuppglossarysubentryfield
instead.)
\ifglxindy
  \newcommand*{\@glossarysubentryfield}{%
    \string\subglossentry}
\else
  \newcommand*{\@glossarysubentryfield}{%
    \string\subglossentry}
\fi

```

```
\@glo@storeentry{\label}
```

\@glo@storeentry

Determine the format to write the entry in the glossary output (.glo) file. The argument is the entry's label (should already have been de-tok'ed if required). The result is stored in \glo@<label>@index, where <label> is the entry's label. (This doesn't include any formatting or location information.)

```
\newcommand{\@glo@storeentry}[1]{%
```

Escape makeindex/xindy special characters in the label:

```
\protected@edef\@glo@esclabel{#1}%  
\@gls@checkmkidxchars\@glo@esclabel
```

Get the sort string and escape any special characters

```
\protected@edef\@glo@sort{\csname glo@#1@sort\endcsname}%  
\@gls@checkmkidxchars\@glo@sort
```

Same again for the name string. Escape any special characters in the prefix

```
\@gls@checkmkidxchars\@glo@prefix
```

Get the parent, if one exists

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

Write the information to the glossary file.

```
\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]{\gls@ifmeasuring{}\{#1\}}

\gls@ifmeasuring
    \newcommand*\gls@ifmeasuring{\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 \@one
        }%
        {}%
    }

\gls@start@measuring It's best to scope this, but still provide some support to switch on and off.
    \newcommand{\gls@start@measuring}{%
        \ifx\gls@ifmeasuring\@firstoftwo
        \else
            \let\gls@org@target\glstarget
            \let\gls@org@link\glslink
            \def\gls@measuring@reset{}%
        \fi
        \let\gls@ifmeasuring\@firstoftwo
        \let\refstepcounter\gls@local@stepcounter
        \let\label\@gobble
        \let\glstarget\@secondoftwo
        \let\glslink\@secondoftwo
    }

\gls@stop@measuring
    \newcommand{\gls@stop@measuring}{%
        \renewcommand*\gls@ifmeasuring{\gls@if@measuring}%
        \let\refstepcounter\gls@org@refstepcounter
        \let\label\gls@org@label
        \let\glstarget\gls@org@target
        \let\glslink\gls@org@link
        \gls@measuring@reset
        \def\gls@measuring@reset{}%
    }

\glsmeasureheight
    \newcommand{\glsmeasureheight}[2]{%
        \settoheight{#1}{\gls@start@measuring#2\gls@stop@measuring}%
    }

\glsmeasuredepth
    \newcommand{\glsmeasuredepth}[2]{%

```



```

        \settodepth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
    }

\glsmeasurewidth
    \newcommand{\glsmeasurewidth}[2]{%
        \settoheight{#1}{\gls@start@measuring#2\gls@stop@measuring}%
    }

\glspatchtabularx Patch \TX@trial (as per David Carlisle's answer in http://tex.stackexchange.com/a/94895). This does nothing if \TX@trial hasn't been defined.
    \def\@gls@patchtabularx#1\hbox#2#3!!{%
        \def\TX@trial##1{#1\hbox{\gls@start@measuring#2\gls@stop@measuring}#3}%
    }
    \newcommand*\glspatchtabularx{%
        \ifdef\TX@trial
        {%
            \expandafter\@gls@patchtabularx\TX@trial{##1}!!%
            \let\glspatchtabularx\relax
        }%
    }%
}

```

\glsreset The command `\glsreset{<label>}` can be used to set the entry flag to indicate that it hasn't been used yet. The required argument is the entry label.

```

\newcommand*\glsreset[1]{%
    \gls@ifnotmeasuring
    {%
        \glsdoifexists{#1}%
        {%
            \@glsreset{#1}%
        }%
    }%
}

```

\glslocalreset As above, but with only a local effect:

```

\newcommand*\glslocalreset[1]{%
    \gls@ifnotmeasuring
    {%
        \glsdoifexists{#1}%
        {%
            \glslocalreset{#1}%
        }%
    }%
}

```

\glsunset The command `\glsunset{<label>}` can be used to set the entry flag to indicate that it has been used. The required argument is the entry label.

```

\newcommand*\glsunset[1]{%
    \gls@ifnotmeasuring

```

```

    {%
      \glsdoifexists{#1}%
      {%
        \@glsunset{#1}%
      }%
    }%
  }

\glslocalunset As above, but with only a local effect:
  \newcommand*{\glslocalunset}[1]{%
    \gls@ifnotmeasuring
    {%
      \glsdoifexists{#1}%
      {%
        \@glslocalunset{#1}%
      }%
    }%
  }

\@glslocalunset Local unset. This defaults to just \@glslocalunset but is changed by
\glsenableentrycount.
  \newcommand*{\@glslocalunset}{\@glslocalunset}

\@@glslocalunset Local unset without checks.
  \newcommand*{\@@glslocalunset}[1]{%
    \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
  }

\@glsunset Global unset. This defaults to just \@@glsunset but is changed by \glsenableentrycount.

  \newcommand*{\@glsunset}{\@@glsunset}

\@@glsunset Global unset without checks.
  \newcommand*{\@@glsunset}[1]{%
    \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
  }

\@glslocalreset Local reset. This defaults to just \@@glslocalreset but is changed by
\glsenableentrycount.
  \newcommand*{\@glslocalreset}{\@@glslocalreset}

\@@glslocalreset Local reset without checks.
  \newcommand*{\@@glslocalreset}[1]{%
    \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
  }

\@glsreset Global reset. This defaults to just \@@glsreset but is changed by \glsenableentrycount.

  \newcommand*{\@glsreset}{\@@glsreset}

```

`\@@glsreset` Global reset without checks.

```
\newcommand*{\@@glsreset}[1]{%
  \expandafter\global\csname glo@\glsdetoklabel{#1}@flagfalse\endcsname
}
```

Reset all entries for the named glossaries (supplied in a comma-separated list). Syntax: `\glsresetall[<glossary-list>]`

`\glsresetall`

```
\newcommand*{\glsresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsreset{\@glsentry}%
  }%
}
```

As above, but with only a local effect:

`\glslocalresetall`

```
\newcommand*{\glslocalresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalreset{\@glsentry}%
  }%
}
```

Unset all entries for the named glossaries (supplied in a comma-separated list). Syntax: `\glsunsetall[<glossary-list>]`

`\glsunsetall`

```
\newcommand*{\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsunset{\@glsentry}%
  }%
}
```

As above, but with only a local effect:

`\glslocalunsetall`

```
\newcommand*{\glslocalunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalunset{\@glsentry}%
  }%
}
```

2.9 Keeping Track of How Many Times an Entry Has Been Unset

Version 4.14 introduced `\glsenableentrycount` that keeps track of how many times an entry is marked as used. The counter is reset back to zero when the first use flag is reset. Note that although the word “counter” is used here, it’s not an actual L^AT_EX counter or even an explicit T_EX count register but is just a macro. Any of the commands that use `\glsunset` or `\glslocalunset`, such as `\gls`, will automatically increment this value. Commands that don’t modify the first use flag (such as `\glstext` or `\glsentrytext`) don’t modify this value.

`\newglossaryentry@defcounters` Define entry fields to keep track of how many times that entry has been marked as used.

```
\newcommand*{\@newglossaryentry@defcounters}{%
  \csdef{glo@\@glo@label @currcount}{0}%
  \csdef{glo@\@glo@label @prevcount}{0}%
}
```

`\ifglsresetcurrcount` A conditional to determine whether or not to reset the entry count to zero if the first use flag is reset. The default is now not to do this, which changes the implementation for v4.49 and below.

```
\newif\ifglsresetcurrcount
\glsresetcurrcountfalse
```

`\glsenableentrycount` Enables tracking of how many times an entry has been marked as used.

```
\newcommand*{\glsenableentrycount}{%
```

Enable new entry fields.

```
\let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
```

Disable `\newglossaryentry` in the document environment.

```
\renewcommand*{\gls@defdocnewglossaryentry}{%
  \renewcommand*{\newglossaryentry[2]}{%
    \PackageError{glossaries}{\string\newglossaryentry\space
      may only be used in the preamble when entry counting has
      been activated}{If you use \string\glsenableentrycount\space
      you must place all entry definitions in the preamble not in
      the document environment}%
  }%
}
```

Define commands `\glsentrycurrcount` and `\glsentryprevcount` to access these new fields. Default to zero if undefined.

```
\newcommand*{\glsentrycurrcount}[1]{%
  \ifcsundef{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 `\cgl`s. (Only global unset is used if previous count was one as it doesn't make sense to have a local unset here given that the previous count was global.)

```
\def\@cgl@s@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cgl@sformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@gls@{##1}{##2}[##3]%
  \fi
}%
```

Similarly for the analogous commands. No case change plural:

```
\def\@cgl spl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cgl splformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@cgl spl@{##1}{##2}[##3]%
  \fi
}%
```

First letter uppercase singular:

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

First letter uppercase plural:

```

\def\cGlspl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cGlsplformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \cGlspl@{##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}%
  }

```

s@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]{}`

`\cgl`s Define command that works like `\gls` but behaves differently if the entry count function is enabled. (If not enabled, it behaves the same as `\gls` but issues a warning.)
`\newrobustcmd*{\cgl}{\@gls@hyp@opt\@cgl}`

`\@cgl`s Defined the un-starred form. Need to determine if there is a final optional argument
`\newcommand*{\@cgl}[2][{}]{%`
`\new@ifnextchar[{\@cgl@{#1}{#2}}{\@cgl@{#1}{#2}[]}]%`
`}`

`\@cgl`s@ Read in the final optional argument. This defaults to same behaviour as `\gls` but issues a warning.
`\def\@cgl@#1#2[#3]{%`
`\GlossariesWarning{\string\cgl\space is defaulting to`
`\string\gls\space since you haven't enabled entry counting}%`
`\@cgl@{#1}{#2}[#3]%`
`}`

`\cgl`sformat Format used by `\cgl`s if entry only used once on previous run. The first argument is the label, the second argument is the insert text.
`\newcommand*{\cglformat}[2]{%`
`\ifglshaslong{#1}{\glentrylong{#1}}{\glentryfirst{#1}}#2%`
`}`

`\cGl`s Define command that works like `\Gls` but behaves differently if the entry count function is enabled. (If not enabled, it behaves the same as `\Gls` but issues a warning.)
`\newrobustcmd*{\cGl}{\@gls@hyp@opt\@cGl}`

Add sentence case mapping.
`\glsmfuaddmap{\cgl}{\cGl}`

`\@cGl`s Defined the un-starred form. Need to determine if there is a final optional argument
`\newcommand*{\@cGl}[2][{}]{%`
`\new@ifnextchar[{\@cGl@{#1}{#2}}{\@cGl@{#1}{#2}[]}]%`
`}`

`\@cGl`s@ Read in the final optional argument. This defaults to same behaviour as `\Gls` but issues a warning.
`\def\@cGl@#1#2[#3]{%`
`\GlossariesWarning{\string\cGl\space is defaulting to`
`\string\Gls\space since you haven't enabled entry counting}%`
`\@cGl@{#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%
}

```

`\cglsp1` Define command that works like `\glsp1` but behaves differently if the entry count function is enabled. (If not enabled, it behaves the same as `\glsp1` but issues a warning.)

```

\newrobustcmd*{\cglsp1}{\@gls@hyp@opt\@cglsp1}

```

`\@cglsp1` Defined the un-starred form. Need to determine if there is a final optional argument

```

\newcommand*{\@cglsp1}[2][{}]{%
  \new@ifnextchar[{\@cglsp1@{#1}{#2}}{\@cglsp1@{#1}{#2}[]}%
}

```

`\@cglsp1@` Read in the final optional argument. This defaults to same behaviour as `\glsp1` but issues a warning.

```

\def\@cglsp1@#1#2[#3]{%
  \GlossariesWarning{\string\cglsp1\space is defaulting to
    \string\glsp1\space since you haven't enabled entry counting}%
  \@glsp1@{#1}{#2}[#3]%
}

```

`\cglsp1format` Format used by `\cglsp1` if entry only used once on previous run. The first argument is the label, the second argument is the insert text.

```

\newcommand*{\cglsp1format}[2]{%
  \ifglshaslong{#1}{\glsp1entrylongpl{#1}}{\glsp1entryfirstplural{#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{\cglsp1}{\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}
```

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”.

¹and any other valid L^AT_EX code that can be used in the preamble.

```

\glstextformat
    \newcommand*{\glstextformat}[1]{#1}

\glentryfmt As from version 3.11a, the way in which an entry is displayed is now governed by
\glentryfmt. 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*{\glentryfmt}{\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
                        \glentryplural{\glslabel}\glsinsert
                        }%
                        {%
                            First use
                            \glentryfirstplural{\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  
}%  
}
```

`\glsngenacfmt` Define a generic acronym format that uses the long and short keys (or their plurals) and `\acrfullformat`, `\firstacronymfont` and `\acronymfont`.

```
\newcommand*{\glsngenacfmt}{%  
  \ifdefempty\glscustomtext  
  {%  
    \ifglsused\glslabel  
    {%
```

Subsequent use:

```
\glsifplural  
{%
```

Subsequent plural form:

```
\glsupercase  
{%
```

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:

```
\glsupercase  
{\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert}%  
}%  
}%  
{%
```

Subsequent singular form

```
\glscapscase  
{%
```

Subsequent singular form, don't adjust case:

```
\acronymfont{\glentryshort{\glslabel}}\glsinsert  
}%  
{%
```

Subsequent singular form, make first letter upper case:

```
\acronymfont{\Glentryshort{\glslabel}}\glsinsert  
}%  
{%
```

Subsequent singular form, all caps:

```
\glsupercase  
{\acronymfont{\glentryshort{\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:

```
\glsupercase  
{\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\glsentencecase\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{<label>}{<insert>}
```

`\Genplacrfullformat`

As above but makes the first letter upper case.

```
\newcommand*{\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}
```

`\glsdisplayfirst` `\glsdisplayfirst` deprecated in v3.11a (2013-10-15) and removed in v4.50. Use rollback if required.

`\glsdisplay` `\glsdisplay` deprecated in v3.11a (2013-10-15) and removed in v4.50. Use rollback if required.

`\defglsdisplay` `\defglsdisplay` deprecated in v3.11a (2013-10-15) and removed in v4.50. Use rollback if required.

`\defglsdisplayfirst` `\defglsdisplayfirst` deprecated in v3.11a (2013-10-15) and removed in v4.50. Use rollback if required.

2.11.1 Links to glossary entries

The links to glossary entries all have a first optional argument that can be used to change the format and counter of the associated entry number. Except for `\glslink` and `\glsdisp`, the commands like `\gls` have a final optional argument that can be used to insert additional text in the link (this will usually be appended, but can be redefined using `\defglsentryfmt`). It goes against the L^AT_EX norm to have an optional argument after the mandatory arguments, but it makes more sense to write, say, `\gls{label}[’s]` rather than, say, `\gls[append=’s]{label}`. Since these control sequences are defined to include the final square bracket, spaces will be ignored after them. This is likely to lead to confusion as most users would not expect, say, `\gls{<label>}` to ignore following spaces, so `\new@ifnextchar` from the `amsgen` package is required.

The following keys can be used in the first optional argument. The `counter` key checks that the value is the name of a valid counter.

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

```
}%
}
```

The value of the `format` key should be the name of a command (without the initial backslash) that has a single mandatory argument which can be used to format the associated entry number.

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

```
s@glslink@hyper@update@hook
```

```
\newcommand*{\@gls@glslink@hyper@update@hook}{}
```

The `hyper` key is a boolean key, it can either have the value `true` or `false`, and indicates whether or not to make a hyperlink to the relevant glossary entry. If `hyper` is `false`, an entry will still be made in the glossary, but the given text won't be a hyperlink.

```
\define@boolkey{glslink}{hyper}[true]{\@gls@glslink@hyper@update@hook}
```

Initialise `hyper` key.

```
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
```

Commands like `\gls` and `\glsdisp` unset the first use flag afterwards. Provide a way to localise this.

```
\@gls@save@glslocal
```

```
\newcommand*{\@gls@save@glslocal}{%
\let\if@org@KV@glslink@local\ifKV@glslink@local
}
```

```
\@gls@restore@glslocal
```

```
\newcommand*{\@gls@restore@glslocal}{%
\ifKV@glslink@local
\let\@gls@do@glsunset@glslocalunset
\else
\let\@gls@do@glsunset@glsunset
\fi
\let\ifKV@glslink@local\if@org@KV@glslink@local
}
```

```
\@gls@do@glsunset
```

```
\newcommand*{\@gls@do@glsunset}[1]{\glsunset{#1}}
```

The `local` key is a boolean key. If `true` this indicates that commands such as `\gls` should only do a local reset rather than a global one.

```
\define@boolkey{glslink}{local}[true]{}
\@gls@save@glslocal
```

The original `\glsifhyper` command isn't particularly useful as it makes more sense to check the actual hyperlink setting rather than testing whether the starred or unstarred version has been used. Therefore, as from version

4.08, `\glsifhyper` is deprecated in favour of `\glsifhyperon`. In case there is a particular need to know whether the starred or unstarred version was used, provide a new command that determines whether the *-version, +-version or unmodified version was used.

```
\glslinkvar{<unmodified case>}{<star case>}{<plus case>}
```

`\glslinkvar` Initialise to unmodified case.

```
\newcommand*{\glslinkvar}[3]{#1}
```

`\glsifhyper` `\glsifhyper` deprecated in v4.08 (2014-07-30) and removed in v4.50.

`\@gls@hyp@opt` Used by the commands such as `\glslink` to determine whether to modify the hyper option.

```
\newcommand*{\@gls@hyp@opt}[1]{%
\let\glslinkvar\@firstofthree
\let\@gls@hyp@opt@cs#1\relax
\@ifstar{\s@gls@hyp@opt}%
{\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
```

`\s@gls@hyp@opt` Starred version

```
\newcommand*{\s@gls@hyp@opt}[1][]{%
\let\glslinkvar\@secondofthree
\@gls@hyp@opt@cs[hyper=false,#1]}
```

`\p@gls@hyp@opt` Plus version

```
\newcommand*{\p@gls@hyp@opt}[1][]{%
\let\glslinkvar\@thirdofthree
\@gls@hyp@opt@cs[hyper=true,#1]}
```

Syntax:

```
\glslink[<options>]{<label>}{<text>}
```

Display `<text>` in the document, and add the entry information for `<label>` into the relevant glossary. The optional argument should be a key value list using the `\glslink` keys defined above.

There is also a starred version:

```
\glslink*[<options>]{<label>}{<text>}
```

which is equivalent to `\glslink[hyper=false,<options>]{<label>}{<text>}`

First determine which version is being used:

```

\glslink
\newrobustcmd*{\glslink}{%
  \@gls@hyp@opt\@gls@@link
}

```

`\@gls@@link` The main part of the business is in `\@gls@link` which shouldn't check if the term is defined as it's called by `\gls` etc which also perform that check.

```

\newcommand*{\@gls@@link}[3][]{%
  \glsdoifexistsordo{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \@gls@link[#1]{#2}{#3}%
  }%

```

Display the specified text. (The entry doesn't exist so there's nothing to link it to.)

```

  \glstextformat{#3}%
}%

\glspostlinkhook
}

```

`\Glslink` Provide a sentence-case version in order to set up a map.

```

\newrobustcmd*{\Glslink}{%
  \@gls@hyp@opt\@Gls@@link
}

\newcommand*{\@Gls@@link}[3][]{%
  \@gls@@link[#1]{#2}{\glsentencecase{#3}}%
}
\glsmfuaddmap{\glslink}{\Glslink}

```

`\glspostlinkhook`

```

\newcommand*{\glspostlinkhook}{}

```

`\@gls@link@checkfirsthyper` Check for first use and switch off hyper key if hyperlink not wanted. (Should be off if first use and `hyper=false` is on or if first use and both the entry is in an acronym list and the `acrfootnote` setting is on.) This assumes the glossary type is stored in `\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@glstdisablehyperinlist Disable hyperlink if in the “nohyper” list.
\newcommand*{\do@glstdisablehyperinlist}{%
\expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
{\KV@glslink@hyperfalse}}%
}

\gls@setdefault@glslink@opts Hook to set default options for \@gls@link.
\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{glolinkprefixglslabel}{\glstextformat{#3}}%
\else
  \glsdonohyperlink{glolinkprefixglslabel}{\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 \@currentenv 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\theHglentrycounter{\@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\theHglentrycounter{\theglsentrycounter}%
}%
{%
\protected@edef\theHglentrycounter{\expandafter\noexpand
\csname theH\@gls@counter\endcsname}%
}%
\fi
```

}

`\@set@glo@numformat` Set the formatting information in the format required by `makeindex`. The first argument is the format specified by the user (via the format key), the second argument is the name of the counter used to indicate the location, the third argument is a control sequence which stores the required format and the fourth argument (new to v3.0) is the hyper-prefix.

```
\def\@set@glo@numformat#1#2#3#4{%
  \expandafter\@glo@check@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{%
```



```

\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\ "\nnull}%
    \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\ ?\?\nnull}%
    \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
```

```

\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
}

```

```

\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}%
      \fi
    \fi
  }
}

```

```

\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}%
    \raisebox{\gls@tmplen}{\hypertarget{#1}{}}%
    #2%
  }%
}

```

\glsdohyperlink

```

\newcommand*{\glsdohyperlink}[2]{%
  \glsdoshowtarget{#1}{\hyperlink{#1}{#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
}

```

`\@glslink` If `\hypertarget` is not defined, `\@glslink` ignores its first argument and just does the second argument, otherwise it is equivalent to `\hypertarget`.

```

\ifcsundef{hypertarget}%
{%
\let\@glslink\@secondoftwo
}%
{%
\let\@glslink\glsdohypertarget
}

```

Glossary hyperlinks can be disabled using `\glsdisablehyper` (effect can be localised):

```

\glsdisablehyper
\newcommand{\glsdisablehyper}{%
\KV@glslink@hyperfalse
\let\@glslink\glsdonohyperlink
\let\@glslink\@secondoftwo
}

```

Glossary hyperlinks can be enabled using `\glsenablehyper` (effect can be localised):

```

\glsenablehyper
\newcommand{\glsenablehyper}{%
\KV@glslink@hypertrue
\let\@glslink\glsdohyperlink
\let\@glslink\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\glsupcase\@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@gl@link@checkfirsthyper\@gl@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 `\@gl@link` sets `\glstype`.

```
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
```

Call `\@gl@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.

```
\@gl@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
\@gl@do@gl@unset{#2}%
}%
```

```
\glspostlinkhook
}
```

`\GLS` behaves like `\gls`, but the link text is converted to uppercase:

`\GLS`

```
\newrobustcmd*{\GLS}{\@gl@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@gl@link@checkfirsthyper\@gl@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@glset{#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
}
```

`\GLSp1` behaves like `\glsp1` except that all the link text is converted to uppercase.

`\GLSp1`

```
\newrobustcmd*{\GLSp1}{\@gls@hyp@opt\@GLSp1}
\glsmfublocker{\GLSp1}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSp1}[2] []{%
  \new@ifnextchar[{\@GLSp1@{#1}{#2}}{\@GLSp1@{#1}{#2} []}%
}
```

`\@GLSp1` Read in the final optional argument:

```
\def\@GLSp1@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper

    \let\glsifplural\@firstoftwo
    \let\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@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
```

```
\let\gl@ifplural\@secondoftwo
\let\gl@scapscase\@firstofthree
\def\glscustomtext{#3}%
\def\glinsert{}%
```

Determine what the link text should be (this is stored in `\@glo@text`) Note that `\@gl@link` sets `\glstype`.

```
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
```

Call `\@gl@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.

```
\@gl@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
\@gl@do@gl@unset{#2}%
}%
\glpostlinkhook
}
```

`\Glsdisp` Provide a sentence-case version to make it easier to map.

```
\newrobustcmd*{\Glsdisp}{\@gl@hyp@opt\@Glsdisp}
```

`\@Glsdisp`

```
\newcommand*{\@Glsdisp}[3][]{%
\@gl@disp[#1]{#2}{\gl@sentencecase{#3}}%
}
\glsmfuaddmap{\gl@disp}{\@Glsdisp}
```

`\@gl@link@nocheckfirsthyper` Instead of just setting `\do@gl@link@checkfirsthyper` to `\relax` in `\@gl@field@link`, set it to `\@gl@link@nocheckfirsthyper` in case some other action needs to take place.

```
\newcommand*{\@gl@link@nocheckfirsthyper}{}%
```

`\@gl@field@link`

```
\@gl@field@link{<options>}{<label>}{<link text>}
```

```
\newcommand{\@gl@field@link}[3]{%
\gl@ifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\@gl@link[#1]{#2}{#3}%
}%
\glpostlinkhook
}
```

`\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}{\glstentrytext{#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}{\glssupercase{\glstentrytext{#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}{\Glstentrytext{#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}{\glstentryfirstplural{#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}{\GLStentryfirstplural{#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}{\glssupercase{\glstentryfirstplural{#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}{\glstentrysymbol{#2}#3}%
}
```

`\Glsymbol` behaves like `\glssymbol` except that the first letter is converted to uppercase.

`\Glsymbol`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glssymbol}{\Glsymbol}
```

Define the un-starred form. Need to determine if there is a final optional argument

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

Read in the final optional argument:

```
\def\@GLssymbol@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\GLstentrysymbol{#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}{\glssupercase{\glstentrysymbol{#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}{\glsentriysymbolplural{#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}{\GLsentriysymbolplural{#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{\glsentriysymbolplural{#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}{\glentryuserii{#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}{\glentryuserii{#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}{\glssupercase{\glentryuserii{#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*{\@GLSserv}[2][\%
\new@ifnextchar[{\@GLSserv@{#1}{#2}}{\@GLSserv@{#1}{#2}[]}]}
```

Read in the final optional argument:

```
\def\@GLSserv@#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*{\@GLSservi}[2] [] {%
  \new@ifnextchar[{\@GLSservi@{#1}{#2}}{\@GLSservi@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSservi@#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@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper

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

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

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

\glspostlinkhook
}
```

\Acrshortpl

```
\newrobustcmd*{\Acrshortpl}{\@gl@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@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper

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

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

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

```

\glspostlinkhook
}

\ACRshortpl
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\ns@ACRshortpl}
\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\gls caps case\@thirdofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \glsupercase{\acronymfont{\glsentryshortpl{#2}}#3}%
    }%

Call \@gls@link Note that \@gls@link sets \gls type.
    \@gls@link[ #1]{ #2}{\csname gls@ \gls type @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\glsupcase\@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\glsupcase\@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\gls caps case\@firstofthree
\let\glsinsert\@empty
```

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

```
\def\gls customtext{%
  \glsentrylongpl{#2}#3%
}%
```

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

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

`\Acr long pl`

```
\newrobustcmd*{\Acr long pl}{\@gls@hyp@opt\ ns@Acr long pl}
```

Add sentence case mapping.

```
\gls mfu add map{\Acr long pl}{\Acr long pl}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\ ns@Acr long pl}[2][ ]{%
  \new@if next char[{\@Acr long pl{#1}{#2}}{\@Acr long pl{#1}{#2}][ ]}%
}
```

Read in the final optional argument:

```
\def\@Acr long pl#1#2[#3]{%
  \gls do if exists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\gls caps case\@secondofthree
    \let\glsinsert\@empty
```

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

```
\def\gls customtext{%
  \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][\@ns@ACRlongpl]{\@ACRlongpl{#1}{#2}[\@ns@ACRlongpl]}
}

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\glsupercase\@thirdofthree
\let\glsinsert\@empty

Bug fix v4.02 removed \acronymfont from \glscustomtext (\acronymfont
only designed for short form).
\def\glscustomtext{%
\glsupercase{\glsentrylongpl{#2}#3}%
}%

Call \@gls@link. Note that \@gls@link sets \glsstyle.
\@gls@link[#1]{#2}{\csname gls@\glsstyle @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{<cs>}{<label>}{<field>}
```

\glsletentryfield

```
\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\glsentencecase\expandafter{\@glo@text}%
    }%
    ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
    entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
    label and the field name}%
  }%
}%
??%
}%
}
```

Aside from \Glsentryname, the \Glsentry... sentence case commands can all follow the same format.

```
\@Glsentryfield{<entry-label>}{<field-label>}
```

\@Glsentryfield

```
\newcommand{\@Glsentryfield}[2]{%
  \glstexorpdfstring{\@Gls@entry@field{#1}{#2}}%
  {\MFUsentencecase{\@gls@entry@field{#1}{#2}}}%
}
```

`\glstexorpdfstring` Provide a command that will use `\texorpdfstring` if it has been defined. (Remove all the cumbersome tests to determine whether or not that command has been defined.)

```
\ifdef\texorpdfstring
{\newcommand{\glstexorpdfstring}{\texorpdfstring}}
{\newcommand{\glstexorpdfstring}[2]{#1}}
```

Get the entry name (as specified by the `name` key when the entry was defined). The argument is the label associated with the entry. Note that unless you used `name=false` in the `sanitize` package option you may get unexpected results if the `name` key contains any commands.

`\glsentryname`

```
\newcommand*{\glsentryname}[1]{\@gls@entry@field{#1}{name}}
```

The sentence-case equivalents were made robust since they use the non-expandable `\makefirstuc`. These will now expand in the PDF bookmarks to sentence case using `\MFUsentencecase` (which is expandable). Obviously, if any of the fields contain non-expandable content (such as acronym font changing commands) then that will cause a problem (which can be dealt with by using `hyperref`'s `\pdfstringdefDisableCommands`).

`\Glsentryname`

```
\newcommand*{\Glsentryname}[1]{%
  \glstexorpdfstring{\protect\@Gls@entryname{#1}}{%
    {\MFUsentencecase{\@gls@entry@field{#1}{name}}}%
  }
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsentryname}{\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\glsentencecase\expandafter{\@glo@text}%
```

```

    }%
    {%
Not an acronym
    \@Gls@entry@field{#1}{name}%
    }%
}

```

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

```

\newcommand*{\@org@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo\@glsetoklabel{#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.

```
\glentrydesc
  \newcommand*\glentrydesc[1]{\@gls@entry@field{#1}{desc}}
```

```
\Glsentrydesc
  \newcommand*\Glsentrydesc[1]{\@Glsentryfield{#1}{desc}}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentrydesc}{\Glsentrydesc}
```

Plural form:

```
\glentrydescplural
  \newcommand*\glentrydescplural[1]{%
    \@gls@entry@field{#1}{descplural}%
  }
```

```
\Glsentrydescplural
  \newcommand*\Glsentrydescplural[1]{%
    \@Glsentryfield{#1}{descplural}%
  }
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentrydescplural}{\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:

```
\glentrytext
  \newcommand*\glentrytext[1]{\@gls@entry@field{#1}{text}}
```

```
\Glsentrytext
  \newcommand*\Glsentrytext[1]{%
    \@Glsentryfield{#1}{text}%
  }
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentrytext}{\Glsentrytext}
```

Get the plural form:

```
\glentryplural
  \newcommand*\glentryplural[1]{%
    \@gls@entry@field{#1}{plural}%
  }
```


`\Glsentryplural`

```
\newcommand*{\Glsentryplural}[1]{%
  \@Glsentryfield{#1}{plural}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsentryplural}{\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}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsentrysymbol}{\Glsentrysymbol}
```

Plural form:

`\glsentrysymbolplural`

```
\newcommand*{\glsentrysymbolplural}[1]{%
  \@gls@entry@field{#1}{symbolplural}%
}
```

`\Glsentrysymbolplural`

```
\newcommand*{\Glsentrysymbolplural}[1]{%
  \@Glsentryfield{#1}{symbolplural}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsentrysymbolplural}{\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}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentryfirst}{\Glentryfirst}
```

Get the plural form (as specified by the `firstplural` key when the entry was defined).

`\glentryfirstplural`

```
\newcommand*{\glentryfirstplural}[1]{%
  \@gls@entry@field{#1}{firstpl}%
}
```

`\Glentryfirstplural`

```
\newcommand*{\Glentryfirstplural}[1]{%
  \@Glentryfield{#1}{firstpl}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentryfirstplural}{\Glentryfirstplural}
```

`\glscapitalisewords` Make it easier to switch from `\capitalisewords` to `\capitalisefmtwords`.

```
\newcommand{\glscapitalisewords}[1]{%
  \capitalisewords{#1}%
}
```

`\glentrytitlecase` Now uses sentence case in PDF bookmark.

```
\newcommand*{\glentrytitlecase}[2]{%
  \glstexorpdfstring
  {\@glentrytitlecase{#1}{#2}}%
  {\MFUsentencecase{\@gls@entry@field{#1}{#2}}}%
}
```

`\@glentrytitlecase`

```
\newrobustcmd*{\@glentrytitlecase}[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.

`\glentrytype`

```
\newcommand*{\glentrytype}[1]{\@gls@entry@field{#1}{type}}
```

Display the sort text used for this entry. Note that the `sort` key is sanitized, 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}%
    }

Add sentence case mapping.
    \glsmfuaddmap{\glsentryuseri}{\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}%
    }

Add sentence case mapping.
    \glsmfuaddmap{\glsentryuserii}{\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.

```
\glsmfuaddmap{\glentryuseriii}{\Glentryuseriii}
```

`\glentryuseriv` 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*{\glentryuseriv}[1]{%
  \@gls@entry@field{#1}{useriv}%
}
```

`\Glentryuseriv`

```
\newcommand*{\Glentryuseriv}[1]{%
  \@Glentryfield{#1}{useriv}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentryuseriv}{\Glentryuseriv}
```

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

`\Glentryuserv`

```
\newcommand*{\Glentryuserv}[1]{%
  \@Glentryfield{#1}{userv}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentryuserv}{\Glentryuserv}
```

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

`\Glentryuservi`

```
\newcommand*{\Glentryuservi}[1]{%
  \@Glentryfield{#1}{uservi}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentryuservi}{\Glentryuservi}
```

`\glentryshort` Get the short key (as specified by the `short` the entry was defined). The argument is the label associated with the entry.

```
\newcommand*{\glentryshort}[1]{\@gls@entry@field{#1}{short}}
```

```

\Glsentryshort
    \newcommand*\Glsentryshort[1]{%
        \@Glsentryfield{#1}{short}%
    }

Add sentence case mapping.
    \glsmfuaddmap{\glsentryshort}{\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}%
    }

Add sentence case mapping.
    \glsmfuaddmap{\glsentryshortpl}{\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}%
    }

Add sentence case mapping.
    \glsmfuaddmap{\glsentrylong}{\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}%
    }

Add sentence case mapping.
    \glsmfuaddmap{\glsentrylongpl}{\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})%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsentryfull}{\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})%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsentryfullpl}{\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}{ \& }

\gls hyperlink Provide a hyperlink to a glossary entry without adding information to the glossary file. The entry needs to be added using a command like \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][\glstrytext{\@glo@label}]{%
\def\@glo@label{#2}%
\@glslink{\glo@linkprefix\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[\langle options \rangle]{\langle label \rangle}

```

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 *\langle options \rangle* only has two keys: `counter` and `format` (the `types` key will be ignored).

`\glsadd`

```

\newrobustcmd*{\glsadd}[2][]{%

```

Need to move to horizontal mode if not already in it, but only if not in preamble.

```

\@gls@adjustmode
\glsdoifexists{#2}%
{%
\def\@glsnumberformat{glsnumberformat}%

\protected@edef\@gls@counter{\csname glo@glsdetoklabel{#2}@counter\endcsname}%
\setkeys{glossadd}{#1}%

```

Store the entry's counter in `\theglsentrycounter`

```

\@gls@saveentrycounter

```

Define sort key if necessary:

```

\@gls@setsort{#2}%

```

This should use `\@do@wrglossary` rather than `\do@wrglossary` since the whole point of `\glsadd` is to add a line to the glossary.

```

\@do@wrglossary{#2}%
}%
}

```

Identify `\glsadd` as an exclusion, but this will only work if the optional argument is omitted. If the optional argument is required, the `\glsadd` command with arguments will need to be encapsulated with `\MFUskipunc`. See the `mfistuc v2.08+` manual for further details.

```

\glsmfuexcl{\glsadd}

```


`\@gls@adjustmode`

```
\newcommand*{\@gls@adjustmode}{}
\AtBeginDocument{\renewcommand*{\@gls@adjustmode}{\ifvmode\mbox{}\fi}}
```

The following aren't added as exclusions, since there's no reason for them to occur at the start of content that may require a case-change (such as at the start of a description).

`\glsaddall[<option list>]`

Add all terms defined for the listed glossaries (without displaying any text). If `types` key is omitted, apply to all glossary types.

`\glsaddall`

```
\newrobustcmd*{\glsaddall}[1][]{%
  \protected@edef\@glo@type{\@glo@types}%
  \setkeys{glossadd}{#1}%
  \forallglsentries[\@glo@type]{\@glo@entry}{%
    \glsadd[#1]{\@glo@entry}%
  }%
}
```

`\glsaddallunused[<glossary types>]`

`\glsaddallunused`

Add all used terms defined for the listed glossaries (without displaying any text). If optional argument is omitted, apply to all glossary types. This should typically go at the end of the document.

```
\newrobustcmd*{\glsaddallunused}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glo@entry}%
  {%
    \ifglsused{\@glo@entry}{\@glsadd[format=glsignore]{\@glo@entry}}%
  }%
}
```

`\glsignore`

```
\newcommand*{\glsignore}[1]{}%
```

2.13 Creating associated files

The `\writeist` command creates the associated customized `.ist` `makeindex` style file. While defining this command, some characters have their catcodes temporarily changed to ensure they get written to the `.ist` file correctly. The `makeindex` actual character (usually `@`) is redefined to be a `?`, to allow internal commands to be written to the glossary file output file.

The special characters are stored in `\@gls@actualchar`, `\@gls@encapchar`, `\@gls@levelchar` and `\@gls@quotechar` to make them easier to use later, but

don't change these values, because the characters are encoded in the command definitions that are used to escape the special characters (which means that the user no longer needs to worry about `makeindex` special characters).

The symbols and numbers label for group headings are hardwired into the `.ist` file as `glsymbols` and `glsnumbers`, the group titles can be translated (so that `\glsymbolsgroupname` replaces `glsymbols` and `\glsnumbersgroupname` replaces `glsnumbers`) using the command `\glsgetgrouptitle` which is defined in `glossary-hypernav`. This is done to prevent any problem characters in `\glsymbolsgroupname` and `\glsnumbersgroupname` from breaking hyperlinks.

`\glsopenbrace` Define `\glsopenbrace` to make it easier to write an opening brace to a file.
`\edef\glsopenbrace{\expandafter\@gobble\string\{}`

`\glsclosebrace` Define `\glsclosebrace` to make it easier to write an opening brace to a file.
`\edef\glsclosebrace{\expandafter\@gobble\string\}}`

`\glsbackslash` Define `\glsbackslash` to make it easier to write a backslash to a file.
`\edef\glsbackslash{\expandafter\@gobble\string\}`

`\glsquote` Define command that makes it easier to write quote marks to a file in the event that the double quote character has been made active.
`\edef\glsquote#1{\string"#1\string"}`

`\glspcentchar` Define `\glspcentchar` to make it easier to write a percent character to a file.
`\edef\glspcentchar{\expandafter\@gobble\string\%}`

`\glstildechar` Define `\glstildechar` to make it easier to write a tilde character to a file.
`\edef\glstildechar{\string~}`

`\@glsfirstletter` Define the first letter to come after the digits 0,...,9. Only required for `xindy`.
`\ifglsxindy`
`\newcommand*\@glsfirstletter}{A}`
`\fi`

`\GlsSetXdyFirstLetterAfterDigits` Sets the first letter to come after the digits 0,...,9. The starred version sanitizes.
`\newcommand*\GlsSetXdyFirstLetterAfterDigits}{%`
`\@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}`
`\ifglsxindy`
`\newcommand*\@GlsSetXdyFirstLetterAfterDigits[1]{%`
`\renewcommand*\@glsfirstletter}{#1}}`
`\newcommand*\s@GlsSetXdyFirstLetterAfterDigits[1]{%`
`\renewcommand*\@glsfirstletter}{#1}%`
`\@onelevel@sanitize\@glsfirstletter`
`}`
`\else`
`\newcommand*\@GlsSetXdyFirstLetterAfterDigits[1]{%`
`\glsnxindywarning\GlsSetXdyFirstLetterAfterDigits}`
`\newcommand*\s@GlsSetXdyFirstLetterAfterDigits}{%`

```

        \@GlsSetXdyFirstLetterAfterDigits
    }
\fi

```

`\@xdynumbergrouporder` Specifies the order of the number group.

```

\ifglxsindy
  \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
}
\ifglxsindy
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
  }
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
    \@onelevel@sanitize\@xdynumbergrouporder
  }
\else
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \glsnxindywarning\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.

```

\ifglxsindy
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \renewcommand*{\@glsminrange}{#1}}
\else
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \glsnxindywarning\GlsSetXdyMinRangeLength}
\fi

```

`\writeist`

```

\ifglxsindy

```

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=\listfilename
```

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 $\{\langle Hprefix \rangle\}\{\langle number \rangle\}$, so need to add all possible combinations of location types.

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

Case where $\langle Hprefix \rangle$ is empty:

```
\protected@write\glswrite{{(define-location-class  
  \string"\@gls@classI\string"^^J\space\space\space  
  (  
    :sep "{{"  
    \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space  
    :sep "}"  
  )  
  ^^J\space\space\space  
  :min-range-length \@glsminrange^^J%  
  )  
}%
```

Nested iteration over all classes:

```
{%
  \@for\@gls@classII:=\@gls@xdy@locationlist\do{%
    \protected@write\glswrite{}\{define-location-class
      \string"\@gls@classII-\@gls@classI\string"
      ^^J\space\space\space
      (
        :sep "{"
        \csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
        :sep "{{"
        \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
        :sep "}"
      )
      ^^J\space\space\space
      :min-range-length \@glsminrange^^J%
    )
  }%
}%
}%
}%
```

User defined location classes (needs checking for new location format).

```
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
```

Cross-reference class. (The unverified option is used as the cross-references are supplied using the list of labels along with the optional argument for `\glsseeformat` which xindy won't recognise.)

```
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
```

Define how cross-references should be displayed. This adds an empty set of braces after the cross-referencing information allowing for the final argument of `\glsseeformat` which gets ignored. (When using `makeindex` this final argument contains the location information which is not required.)

```
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\glsseeformat\string"
  :close \string"{}\string")}%
```

Provide hook to write extra material here (used by `glossaries-extra` to define a `seealso` class).

```
\@xdycrossrefhook
```

List the order to sort the classes.

```
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xdylocationclassorder))}%
```

Specify what to write to the start and end of the glossary file.

```
\write\glswrite{^^J; define the glossary markup^^J}%
```

```

\write\glswrite{(markup-index^^J\space\space\space
:open \string"\string
\glossarysection[\string\glossarytoctitle]{\string
\glossarytitle}\string\glossarypreamble}%

```

Add all the xindy-only macro definitions (needed to prevent errors in the event that the user changes from xindy to makeindex)

```

\@for\@this@ctr:=\@xdycounters\do{%
{%
\@for\@this@attr:=\@xdyattributelist\do{%
\protected\write\glswrite{}\{\string\providecommand*%
\expandafter\string
\csname glsX\@this@ctr X\@this@attr\endcsname[2]%
{%
\string\setentrycounter
[\expandafter\@gobble\string\#1]{\@this@ctr}%
\expandafter\string
\csname\@this@attr\endcsname
{\expandafter\@gobble\string\#2}%
}%
}%
}%
}%

```

Add the end part of the open tag and the rest of the markup-index information:

```

\write\glswrite{%
\string\begin
{theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
\space\space:close \string"\glpercentchar\glstildechar n\string
\end{theglossary}\string\glossarypostamble
\glstildechar n\string" ^^J\space\space\space
:tree)}}%

```

Specify what to put between letter groups

```

\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\glstildechar n\string"}}%

```

Specify what to put between entries

```

\write\glswrite{(markup-indexentry
:open \string"\string\relax \string\glsresetentrylist
\glstildechar n\string"}}%

```

Specify how to format entries

```

\write\glswrite{(markup-locclass-list :open
\string"\glsopenbrace\string\glossaryentrynumbers
\glsopenbrace\string\relax\space \string"^^J\space\space\space
:sep \string", \string"
:close \string"\glsclosebrace\glsclosebrace\string"}}%

```

Specify how to separate location numbers

```

\write\glswrite{(markup-locoref-list

```

```
:sep \string\string\delimN\space\string"))}%
```

Specify how to indicate location ranges

```
\write\glswrite{(markup-range
:sep \string\string\delimR\space\string"))}%
```

Specify 2-page and 3-page suffixes, if defined. First, the values must be sanitized to write them explicitly.

```
\@onelevel@sanitize\gls@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{\glpercentchar\space makeindex style file
created by the glossaries package}
\write\glswrite{\glpercentchar\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\glsgroupresetentrylist \string}
\write\glswrite{symhead_positive \string\glssymbols\string}
\write\glswrite{numhead_positive \string\glssymbols\string}
\write\glswrite{page_compositor \string\glscpositor\string}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{suffix_2p \string\gls@suffixF\string}
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{suffix_3p \string\gls@suffixFF\string}
\fi

```

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}}
\@onlypremake\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.

```

\ifglxsindy
\newcommand*\GlsSetQuote[1]{\glsglsmakeindexwarning\GlsSetQuote}
\newcommand*\Gls@nosetquote[1]{\glsglsmakeindexwarning\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
    }%
  }%
\noexpand\@gls@docheckquotedef
}
\noexpand\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}}
\noexpand\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-existent file). The relevant glossary must be defined prior to using `\makeglossary`. `glossaries-extra` allows for a hybrid approach.

`\@makeglossary` Unstarred form of `\ifglossaryexists` is used as `\makeglossary` can't be used with an ignored glossary.

```
\newcommand*{\@makeglossary}[1]{%
  \ifglossaryexists{#1}%
  {%
```

Only create a new write if `savewrites=false` otherwise create a token to collect the information.

```
  \ifglssavewrites
    \expandafter\newtoks\csname glo@#1@filetok\endcsname
  \else
    \expandafter\newwrite\csname glo@#1@file\endcsname
    \expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
  \fi
  \@gls@renewglossary
  \writeist
}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘#1’ not defined}%
  {New glossaries must be defined before using \string\makeglossaries}%
}%
```

```

    }

\@glsopenfile Open write file associated with the given glossary.
    \newcommand*{\@glsopenfile}[2]{%
        \immediate\openout#1=\jobname.\csname @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.
    \ifglxindy
        \newcommand*{\@gls@automake}[1]{%
            \ifglossaryexists{#1}
            {%
                \@closegls{#1}%
                \ifdefstring{glsorder}{letter}%
                {\def\@gls@order{-M ord/letorder }}%
                {\let\@gls@order\@empty}%
                \gls@xdy@set@lang{#1}%

                \ifdefvoid\gls@codepage
                {\let\@gls@codepage\@empty}%
                {\def\@gls@codepage{-C \gls@codepage\space}}%
                \edef\@gls@dothiswrite{\noexpand\write18{xindy
                    -I xindy
                    \@gls@order
                    -L \@gls@langmod\space
                    -M \gls@istfilebase\space
                    \@gls@codepage
                    -t \jobname.\csuse{@glotype@#1@log}
                    -o \jobname.\csuse{@glotype@#1@in}
                    \jobname.\csuse{@glotype@#1@out}}}%
               }%
                \@gls@dothiswrite
            }%
            {%
                \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
            }%
        }
    \else
        \newcommand*{\@gls@automake}[1]{%
            \ifglossaryexists{#1}
            {%
                \@closegls{#1}%

```

```

\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-1 }}%
{\let\@gls@order\@empty}%
\edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
-s \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.

```

\ifglxindy
\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}{%
\ifglxindy
\def\glsindexingsetting{xindy}%
\else
\def\glsindexingsetting{makeindex}%
\fi
\@domakeglossaries
}%

```

If `automake=immediate` setting is on, use the shell escape now.

```
\@gls@@automake@immediate
```

Define the `write` used for style file also used for all other output files if `savewrites=true`.

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

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

```

\protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{} }
\protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{} }

```

If `\@gls@extramakeindexopts` has been defined, write it:

```

\ifundef\@gls@extramakeindexopts
{%
}%
\protected@write\@auxout{}{\string\providecommand
\string\@gls@extramakeindexopts[1]{} }
\protected@write\@auxout{}{\string\@gls@extramakeindexopts
{\@gls@extramakeindexopts}}%
}%

```

Write the name of the style file to the aux file (needed by `makeglossaries`)

```
\protected@write\@auxout{}\string\@istfilename{\istfilename}}%
\protected@write\@auxout{}\string\@glsorder{\glsorder}}
```

Iterate through each glossary type and activate it.

```
\@for\@glo@type:=\@glo@types\do{%
  \ifthenelse{\equal{\@glo@type}{}}{\}{%
    \@makeglossary{\@glo@type}}%
}%
```

New glossaries must be created before `\makeglossaries` so disable `\newglossary`.

```
\renewcommand*\newglossary[4][]{%
\PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%
```

Any subsequent instances of this command should have no effect. The deprecated `\makeglossary` is not redefined here as it either implements `\makeglossaries` or has been restored to its original definition (in which case it shouldn't be changed).

```
\let\@makeglossary\@gobble
\let\makeglossaries\relax
```

Disable all commands that have no effect after `\makeglossaries`

```
\@disable@onlypremakeg
```

Allow see key:

```
\let\gls@checkseeallowed\relax
```

Suppress warning about no `\makeglossaries`

```
\let\warn@nomakeglossaries\relax
```

Activate warning about missing `\printglossary`

```
\let\warn@noprntglossary\@makeglossaries@warn@noprntglossary
```

Declare list parser for `\glsdisplaynumberlist`

```
\ifglssavenumberlist
\edef\@gls@doddeflistparser{\noexpand\DeclareListParser
{\noexpand\glsnumlistparser}{\delimN}}%
\@gls@doddeflistparser
\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:

```
\ifglautomake
\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@noprntglossary{%
  \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{<label>}{<handler>}{<xr handler cs>}
```

\glsnumberlistloop

```
\newcommand*{\glsnumberlistloop}[3]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
```

\glsnoidxnumberlistloophandler Handler macro for \glsnumberlistloop. (The argument should be in the form \glsnoidxdisplayloc{<prefix>}{<counter>}{<format>}{<n>})

```
\newcommand*{\glsnoidxnumberlistloophandler}[1]{%
  #1%
}
```

\@no@makeglossaries Can't use both \makeglossaries and \makenoidxglossaries

```
\newcommand*{\@no@makeglossaries}{%
  \PackageError{glossaries}{You can't use both
    \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
    together.}%
}
```

\@gls@noref@warn Warning when no instances of \@gls@reference found.

```
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
    is required to make \string\printnoidxglossary[type={#1}] work}%
}
```

2.14 Writing information to associated files

\gls@noidxglossary Write the glossary information to the aux file (for the ‘noidx’ method).

```
\newcommand*{\gls@noidxglossary}{%
  \protected@write\@auxout{\gls@wrglossdisablelocationcmds}{%
    \string\@gls@reference
      {\csname glo@\@gls@label @type\endcsname}%
      {\@gls@label}%
      {\string\glsnoidxdisplayloc
        {\@glo@counterprefix}%
        {\@gls@counter}%
        {\@glsnumberformat}%
        {\@gls@locref}}%
      }%
  }%
}
```

\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.

```
\forall glossaries{\@glo@type}{%
```

Check for empty glossaries (patch provided by Patrick Häcker)

```
\ifcsundef{glo@\@glo@type @filetok}%
{%
  \def\gls@tmp{}%
}%
{%
  \edef\gls@tmp{\expandafter\the
    \csname glo@\@glo@type @filetok\endcsname}%
}%
\ifx\gls@tmp\@empty
  \ifx\@glo@type\glsdefaulttype
    \GlossariesWarningNoLine{Glossary ‘\@glo@type’ has no
      entries.^^JRemember to use package option ‘nomain’ if
you
      don’t want to^^Juse the main glossary}%
  \else
    \GlossariesWarningNoLine{Glossary ‘\@glo@type’ has no
      entries}%
  \fi
\else
  \@glsopenfile{\glswrite}{\@glo@type}%
  \immediate\write\glswrite{%
    \expandafter\the
      \csname glo@\@glo@type @filetok\endcsname}%
  \immediate\closeout\glswrite
\fi
}%
}
```

As from v4.10, the `\glossary` command isn’t used by the `glossaries` package. Since the user isn’t expected to use this command (as `glossaries` takes care of the particular format required for `makeindex/xindy`) there’s no need for a user level command. Using a custom internal command prevents any conflict with other packages (and with the `\mark` mechanism).

The associated number should be stored in `\theglsentrycounter` before using `\gls@glossary`.

```
\gls@glossary
```

```
\newcommand*{\gls@glossary}[1]{%
```



```
\@gls@glossary{#1}%
}
```

```
\@gls@glossary{<type>}{<indexing info>}
```

`\@gls@glossary`

(In v4.10, `\@glossary` was redefined to `\@gls@glossary` to avoid conflict with other packages.) Initially define internal `\@gls@glossary` to ignore its argument. Indexing will be enabled when `\@gls@glossary` is redefined by `\makeglossary`.

This command was originally defined to do `\@index{<indexing info>}` so that it behaved much like `\index`. The definition was then changed to use `\index` as `memoir` changes the definition of `\@index`. (Thanks to Dan Luecking for pointing this out.)

However, if normal indexing is enabled (for example with `\makeindex`) but no glossary lists are required (so `\makeglossary` isn't used), then `\index` will cause a problem here. The `\@index` trick allows for special characters within `<indexing info>` (so you can do, for example, `\index{%@}%`), and the original design of `\@glossary` here was actually a legacy from the old `glossary` package. With the `glossaries` package, the indexing information supplied in the second argument is more constrained and just consists of the sort value (given by the sort key), the actual value (given by `\glossentry{<label>}` or `\subglossentry{<level>}{<label>}`), and the format. This means that there's no need to worry about special characters appearing in the second argument as they can't be in the label or sort value. (If they are in the sort value then the category code would've needed to be changed when the entry was defined or `\glsperscentchar` 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]{%
  \ifgls@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\beginingroup\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 escloca-
tions=true.
\newcommand*{\gls@disablepagerefexpansion}{%
  \gls@wrglossdisablelocationcmds
  \ifglsesclocations
    \for\@gls@this:=\gls@protected@pagefmts\do

```

```

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

\swrglossdisablelocationcmds

\newcommand{\glswrglossdisablelocationcmds}{}

\gls@alphpage
\newcommand*{\gls@alphpage}{\@alph\c@page}

\gls@Alphpage
\newcommand*{\gls@Alphpage}{\@Alph\c@page}

\gls@numberpage
\newcommand*{\gls@numberpage}{\number\c@page}

\gls@arabicpage
\newcommand*{\gls@arabicpage}{\@arabic\c@page}

\gls@romanpage
\newcommand*{\gls@romanpage}{\romannumeral\c@page}

\gls@Romanpage
\newcommand*{\gls@Romanpage}{\@Roman\c@page}

```

```
\glsaddprotectedpagefmt{<cs name>}
```

```
\glsaddprotectedpagefmt
```

Added a page format to the list of protected page formats. The argument should be the name (without a backslash) of the command that takes a \TeX register as the argument ($\backslash<csname>\c@page$ must be valid).

```

\newcommand*{\glsaddprotectedpagefmt}[1]{%
  \ifglsclocations
    \protected@eappto\gls@protected@pagefmts{\expandonce{\csname gls#1page\endcsname}}%
    \csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
    \protected@eappto\@wrglossarynumberhook{%
      \noexpand\let\expandonce{\csname org@glgls#1\endcsname}%
      \expandonce{\csname#1\endcsname}%
      \noexpand\def\expandonce{\csname#1\endcsname}{%
        \noexpand\@wrglossary@pageformat
        \expandonce{\csname gls#1page\endcsname}%
        \expandonce{\csname org@glgls#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]{%
      \ifglsecllocations
        \@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{\theHglentrycounter}%
      Find the prefix if \@glsHlocref and \@glslocref aren't the same.
      \ifx\@glsHlocref\@glslocref
        \def\@glo@counterprefix{}%
      \else

```

```
\xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
\fi
```

\protected@edef\@gls@label{\glsdetoklabel{#1}}%

```
\@@do@@wrglossary
}
```

```
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodsfalse
```

1. all backslashes in the location must be escaped;
2. `\c@page` can't be prematurely expanded.

For example, suppose `\thepage` is defined as

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⟩}`.

205

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{\theHglentrycounter}%
\endgroup

```

Escape any special characters. It's possible that with `makeindex` the separator might be a `makeindex` special character. Although not likely, it still needs to be taken into account.

```
\@gls@checkmkidxchars\@glslocref
```

Check if the hyper-location is the same as the location and set the hyper prefix.

```

\expandafter\ifx\theHglentrycounter\theHglentrycounter\relax
  \def\@glo@counterprefix{%
\else
  \protected@edef\@glsHlocref{\theHglentrycounter}%
  \@gls@checkmkidxchars\@glsHlocref
  \xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
\fi

```

De-tok label if required

```
\protected@edef\@gls@label{\glsdetoklabel{#1}}%
```

Write the information to file:

```

\@do@wrglossary
}

```

\@@do@@wrglossary

```
\newcommand*{\@@do@@wrglossary}{%
```

Determine whether to use xindy or makeindex syntax

```
\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)

:locoref \string"\@glo@counterprefix}{\@glslocoref}\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}{\@glslocoref}}%
\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 `[\langle tag \rangle]{\langle list \rangle}`, where `\langle tag \rangle` is a tag such as “see” and `\langle list \rangle` is a list of labels.

```
\newcommand{\@do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref
\ifglsxindy
\gls@glossary{\csname glo@#1@type\endcsname}{%
(indexentry
:tkey (\csname glo@#1@index\endcsname)
:xref (\string"\@gls@xref\string")
:attr \string"see\string"
)
}%
}
```

```

\else
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    \string\glossaryentry{\csname glo@#1@index\endcsname
      \@gls@encapchar glsseeformat\@gls@xref}{Z}}%
\fi
}

```

`\@gls@fixbraces` If no optional argument is specified, list needs to be enclosed in a set of braces.

```

\def\@gls@fixbraces#1#2#3\@nil{%
  \ifx#2[\relax
    \@gls@fixbraces#1#2#3\@end@fixbraces
  \else
    \def#1{{#2#3}}%
  \fi
}

```

`\@@gls@fixbraces`

```

\def\@@gls@fixbraces#1[#2]#3\@end@fixbraces{%
  \def#1{[#2]{#3}}%
}

```

```

\glssee \glssee{<label>}{<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 `\glstentryname` (originally in `\glsseeitem`) with `\glstentrytext` to avoid problems with the `name` key being sanitized. This ceased being a problem in v3.08a, but `\glstentrytext` was retained for backward-compatibility. As from v4.50, the default has reverted back to using `\glstentryname` unless the entry is an acronym (which could have a potentially long name).

```
\newcommand*{\glsseeitemformat}[1]{%
```

```
\ifglshashshort{#1}{\glstentrytext{#1}}{\glstentryname{#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{gls@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}}%
}%
}
```

`ss@checkexists@allowignored` For use with `\printunsrtglossary`.

```
\newcommand*{\@printgloss@checkexists@allowignored}[2]{%
\s@ifglossaryexists{#1}%
{#2}%
{\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}
```

`\@printgloss@checkexists`

```
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
```

`\printglossary` has an optional argument. The default value is to set the glossary type to the main glossary.

```
\newcommand*{\printglossary}[1][type=glndefaultttype]{%
\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=glndefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@printnoidx@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{\glndefaulttype}%
  \def\glossarytitle{\csname @glo@type\@glo@type @title\endcsname}%

  \def\glossarytoctitle{\glossarytitle}%
  \let\org@glossarytitle\glossarytitle

  \def\@glossarystyle{%
    \ifx\@glossary@default@style\relax
      \GlossariesWarning{No default glossary style provided \MessageBreak
        for the glossary '\@glo@type'. \MessageBreak
    }
```

```

        Using fallback. \MessageBreak
        To fix this set the style with \MessageBreak
        \string\setglossarystyle\space or use the \MessageBreak
        style key=value option}%
    \fi
}%
\def\gls@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
\bgroup

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

```
\egroup
```

Reset `\glossaryentrynumbers`

```
\global\let\glossaryentrynumbers\@org@glossaryentrynumbers
```

Suppress warning about no `\printglossary`

```
\global\let\warn@noprntglossary\relax
}
```

`\@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`.

```
\ifglxindy
```

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}%  
}%  
}
```

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?

```

\ifinlistcs{\@this@label}{@glsref@#2}%
{%
  \listcsxadd{@glsref@#2}{\@this@label}%
}%
\ifcsdef{@glo@sortingchildren@\@this@label}%
{%
  \@glo@addchildren{#2}{\@this@label}%
}%
}%
}

```

\@glo@addchildren

```
\@glo@addchildren{<type>}{<parent>}
```

```
\newcommand*{\@glo@addchildren}[2]{%
```

Scope to allow nesting.

```

\bgroup
\letcs{\@glo@childlist}{@glo@sortingchildren@#2}%
\@for\@this@childlabel:=\@glo@childlist\do
{%

```

Check this label hasn't already been added.

```

\ifinlistcs{\@this@childlabel}{@glsref@#1}%
{%
  \listcsxadd{@glsref@#1}{\@this@childlabel}%
}%

```

Does this child have children?

```

\ifcsdef{@glo@sortingchildren@\@this@childlabel}%
{%
  \@glo@addchildren{#1}{\@this@childlabel}%
}%
}%
\egroup
}

```

\@glo@do@sortentries

```
\newcommand*{\@glo@do@sortentries}[1]{%
```

```

\ifglshasparent{#1}%
{%
This entry has a parent, so add it to the child list
\protected@edef\@glo@parent{\csuse{glo@\glsdetoklabel{#1}@parent}}%
\ifcsundef{@glo@sortingchildren@\@glo@parent}%
{%
\csdef{@glo@sortingchildren@\@glo@parent}{}%
}%
}%%
\expandafter\@glo@sortedinsert
\csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
Has the parent been added?
\xifinlistcs{\@glo@parent}{\@glsref@\@glo@type}%
{%
Yes, it has so do nothing.
}%
{%
No, it hasn't so add it now.
\expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
}%
}%%
\@glo@sortedinsert{\@glo@sortinglist}{#1}%
}%%
}

```

\@glo@sortedinsert

\@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\dtlcompare{\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 `\glo@list@<type>`.

```

\newcommand*{\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{}%
  \for@gl@sentries[#1]{\@gl@thislabel}%
  {%
    \xifinlistcs{\@gl@thislabel}{\@gl@sref@#1}%
    {%
      \listcsadd{\@glo@sortinglist}{\@gl@thislabel}%
    }%
    {%

```

Hasn’t been referenced.

```

    }%
  }%
  \cslet{\@gl@sref@#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}{\@gl@sref@\@glo@type}%
  {}%
  {%
    \listcsadd{\@gl@sref@\@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@glstetoklabel{#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}%
}%
}
```

`\glsnoidxdisplayloclisthandler` Handler for location list iterator when used with `\glsdisplaynumberlist`.

```
\newcommand*\glsnoidxdisplayloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
```

Same as previous location so skip.

```
}%
}%
  \@gls@noidxloclist@sep
  \@gls@noidxloclist@prev
  \def\@gls@noidxloclist@prev{#1}%
}%
}
```

```
\glsnoidxdisplayloc{<prefix>}{<counter>}{<format>}{<location>}
```

`\glsnoidxdisplayloc`

Display a location in the location list.

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

```
\@gls@reference{<type>}{<label>}{<loc>}
```

`\@gls@reference`

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

```
\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@dotoc@title\relax
}

```

The `toctitle` sets the text used for the relevant entry in the table of contents.

```

\define@key{printgloss}{toctitle}{%
\def@glossarytoctitle{#1}%
\let@gls@dotoc@title\relax
}

```

The `style` key sets the glossary style (but only for the given glossary).

```

\define@key{printgloss}{style}{%
\ifcsundef{@glsstyle@#1}%
{%
\PackageError{glossaries}%
{Glossary style ‘#1’ undefined}{}%
}%
{%
\def@glossarystyle{\setglossarystyle{#1}}%
}%
}

```

The `numberedsection` key determines if this glossary should be in a numbered section.

```

\define@choicekey{printgloss}{numberedsection}%
[\gls@numberedsection@val@gls@numberedsection@nr]%
{false,nolabel,autolabel,nameref}[nolabel]%
{%
\ifcase@gls@numberedsection@nr\relax
\renewcommand*{\@@glossarysecstar}{*}%
\renewcommand*{\@@glossaryseclabel}{}%

```

```

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

```

The `nogroupskip` key determines whether or not there should be a vertical gap between glossary groups.

```

\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
\csuse{glsnogroupskip#1}%
}

```

The `nopostdot` key has the same effect as the package option of the same name.

```

\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
\csuse{glsnopostdot#1}%
}

```

`\GlsEntryCounterLabelPrefix` Make it easier to redefine the label prefix.

```

\newcommand*{\GlsEntryCounterLabelPrefix}{glsentry-}

```

The conditionals have been moved inside the appropriate commands to make it easier for the user to redefine them in the preamble and selectively switch the counter display on and off. Previously the helper commands were redefined by the `entrycounter` option, which would counteract any earlier customisation.

The `entrycounter` key is the same as the package option but localised to the current glossary.

```

\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
\csuse{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]{%
\ifglsonumberlist
  \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 sub-counter can pick up the main counter as the master if required):

```
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
```

`\glsresetsubentrycounter` Resets the `glossarysubentry` counter.

```
\newcommand*{\glsresetsubentrycounter}{%
  \ifglssubentrycounter
    \setcounter{glossarysubentry}{0}%
  \fi
}
```

`\glsresetentrycounter` Resets the `glossaryentry` counter.

```
\newcommand*{\glsresetentrycounter}{%
  \ifglentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
```

`\glsstepentry` Advance the `glossaryentry` counter if in use. The argument is the label associated with the entry.

```
\newcommand*{\glsstepentry}[1]{%
  \ifglentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
```

`\glsstepsubentry` Advance the `glossarysubentry` counter if in use. The argument is the label associated with the subentry.

```
\newcommand*{\glsstepsubentry}[1]{%
  \ifglssubentrycounter

    \protected@edef\currentglssubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
  \fi
}
```

`\glsrefentry` Reference the entry or sub-entry counter if in use, otherwise just do `\gls`.

```
\newcommand*{\glsrefentry}[1]{%
  \ifglentrycounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else
    \ifglssubentrycounter
      \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \fi
  \fi
}
```

```

        \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{\glo@linkprefix#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{<level>}{<label>}{<page-list>}`

`\compatiblesubglossentry`

Removed in v4.50

`\setglossentrycompatibility`

`\glossaryentryfield{<label>}{<name>}{<description>}{<symbol>}{<page-list>}`

`\glossaryentryfield`

This command formerly governed how each entry row should be formatted in the glossary. Now deprecated. Removed in v4.50

`\glossarysubentryfield`

```
\glossarysubentryfield{<level>}{<label>}{<name>}{<description>}{<symbol>}{<page-list>}
```

This command governs how each subentry should be formatted in the glossary. Glossary styles need to redefine this command. Most of the predefined styles ignore *<symbol>*. The first argument is a number indicating the level. (The level should be greater than or equal to 1.)

Removed in v4.50.

`\subglossentry`

```
\subglossentry{<level>}{<label>}{<page-list>}
```

```
\newcommand{\subglossentry}[3]{\glossentryname{#2} #3}
```

Within each glossary, the entries form distinct groups which are determined by the first character of the *sort* key. When using `makeindex`, there will be a maximum of 28 groups: symbols, numbers, and the 26 alphabetical groups A, ..., Z. If you use `xindy` the groups will depend on whatever alphabet is used. This is determined by the language or custom alphabets can be created in the `xindy` style file. The command `\glsgroupskip` specifies what to do between glossary groups. Glossary styles must redefine this command. (Note that `\glsgroupskip` only occurs between groups, not at the start or end of the glossary.)

`\glsgroupskip`

```
\newcommand*{\glsgroupskip}{}%
```

Each of the 28 glossary groups described above is preceded by a group heading. This is formatted by the command `\glsgroupheading` which takes one argument which is the *label* assigned to that group (not the title). The corresponding labels are: `glsymbols`, `glsnumbers`, A, ..., Z. Glossary styles must redefine 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]{%
Even if the argument appears to be a single letter, it won't be considered a
single letter by \dtl@ifsingle if it's an active character.
  \def#2{#1}%
  \regex_match:nnT { ~\A(.|glsnumbers|glssymbols)\Z } { #1 }
  { \ifcsundef{#1groupname}{ }\letcs#2{#1groupname} } }
}
\ExplSyntaxOff
```

`\@gls@noidx@getgrouptitle` Version for the no-indexing app option:

```
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%
  \DTLifint{#1}%
  {\edef#2{\char#1\relax}}%
  {%
    \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
  }%
}
```

`\glsgetgrouplabel{<title>}`

This command does the reverse to the previous command. The argument is the group title, and it produces the group label. Note that if you redefine `\glsgetgrouptitle`, you will also need to redefine `\glsgetgrouplabel`.

`\glsgetgrouplabel` This only works for the symbol and number groups, so is best avoided and not documented. However, it has been converted to use L^AT_EX3 commands.

```

\ExplSyntaxOn
\newcommand*{\glsgetgrouplabel}[1]{%
\str_if_eq:eeTF {#1} { \glssymbolsgroupname }
{ glssymbols }
{
\str_if_eq:eeTF {#1} { \glsnumbersgroupname }
{ glsnumbers }
{ #1 }
}
}
\ExplSyntaxOff

```

The command `\setentrycounter` sets the entry’s associated counter (required by `\glshypernumber` etc.) `\glslink` and `\glsadd` encode the `\glossary` argument so that the relevant counter is set prior to the formatting command.

`\setentrycounter` The prefix is an optional argument because this command originally only had one argument. The optional argument was added in v3.0.

```

\newcommand*{\setentrycounter}[2][{}]{%
\def\@glo@counterprefix{#1}%
\ifx\@glo@counterprefix\@empty
\def\@glo@counterprefix{.}%
\else
\def\@glo@counterprefix{.#1.}%
\fi
\def\glsentrycounter{#2}%
}

```

The current glossary style can be set using `\setglossarystyle{<style>}`.

`\setglossarystyle`

```

\newcommand*{\setglossarystyle}[1]{%
\ifcsundef{@glsstyle@#1}%
{%
\PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
}%
{%
\csname @glsstyle@#1\endcsname
}%
}

```

Set the default style if it’s not already set.

```

\ifx\@glossary@default@style\relax
\protected@edef\@glossary@default@style{#1}%
\fi
}

```

`\glossarystyle` Deprecated in v3.08a. Removed in v4.50.

`\newglossarystyle` New glossary styles can be defined using:

```
\newglossarystyle{<name>}{<definition>}
```

The *<definition>* argument should redefine `\theglossary`, `\glossaryheader`, `\glsgroupheading`, `\glossaryentryfield` and `\glsgroupskip` (see [subsection 2.19](#) for the definitions of predefined styles). Glossary styles should not redefine `\glossarypreamble` and `\glossarypostamble`, as the user should be able to switch between styles without affecting the pre- and postambles.

```
\newcommand{\newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
```

`\renewglossarystyle` Code for this macro supplied by Marco Daniel.

```
\newcommand{\renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
```

Glossary entries are encoded so that the second argument to `\glossaryentryfield` is always specified as `\glsnamefont{<name>}`. This allows the user to change the font used to display the name term without having to redefine `\glossaryentryfield`. The default uses the surrounding font, so in the list type styles (which place the name in the optional argument to `\item`) the name will appear in bold.

`\glsnamefont`

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

Each glossary entry has an associated number list (usually page numbers) that indicate where in the document the entry has been used. The format for these number lists can be changed using the `format` key in commands like `\glslink`. The default format is given by `\glsnumber`. This takes a single argument which may be a single number, a number range or a number list. The number ranges are delimited with `\delimR`, the number lists are delimited with `\delimN`.

If the document doesn’t have hyperlinks, the numbers can be displayed just as they are, but if the document supports hyperlinks, the numbers should link

to the relevant location. This means extracting the individual numbers from the list or ranges. The `hyperref` package does this with the `\hyperpage` command, but this is encoded for comma and dash delimiters and only for the page counter, but this code needs to be more general. So I have adapted the code used in the `hyperref` package.

```
\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{\glstentrycounter\@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[\label]{\langle abbrv \rangle}{\langle long \rangle}{\langle key-val list \rangle}

```

This emulates the way the old `glossary` package defined acronyms. It is equivalent to `\newacronym[\langle key-val list \rangle]{\langle label \rangle}{\langle abbrv \rangle}{\langle long \rangle}` and it additionally defines the command `\langle label \rangle` which is equivalent to `\gls{\langle label \rangle}` (thus `\langle label \rangle` must only contain alphabetical characters). If `\langle label \rangle` is omitted, `\langle abbrv \rangle` 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 `\langle label \rangle` can't have an optional argument if the `xspace` package is loaded. If `xspace` hasn't been loaded then you can do `\langle label \rangle[\langle insert \rangle]` 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 `\glentryfull` 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}{%
    \glssupercase{\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}{%
    \glssupercase{\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

```
\acronymfont{\acronymfont{\glsentryshort{#1}}}
```

Default sort format for acronyms.

```
\newcommand*{\acronymfont}[2]{#1}
```

\setacronymstyle

```
\setacronymfont{\acronymfont{\glsentryshort{#1}}}
```

```
\newcommand*{\setacronymfont}[1]{%
\ifcsundef{@glsacr@dispstyle@#1}%
{%
\PackageError{glossaries}{Undefined acronym style ‘#1’}{}%
}%
}%
\ifdefempty{\@glsacronymlists}%
{%
\DeclareAcronymList{\acronymfont}%
}%
}%
\SetGenericNewAcronym
\GlsUseAcrStyleDefs{#1}%
\@for\@gls@type:=\@glsacronymlists\do{%
\defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{#1}}%
}%
}%
}
```

\newacronymstyle

```
\newacronymfont{\acronymfont{\glsentryshort{#1}}}
```

Defines a new acronym style called *style name*.

```
\newcommand*{\newacronymfont}[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` $\langle long \rangle$ ($\langle short \rangle$) acronym style.

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

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

```
\ifglshaslong{\glslabel}{\glsacrfmt}{\glsacrentryfmt}%
}%
{%
  \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}{\glsгенacfmt}{\glsгенentryfmt}%
}%
{%
\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}{\glsгенacfmt}{\glsгенentryfmt}%
}%
{%
```

```

\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
  \protect\firstacronymfont{\glsenentryshort{##1}}##2\space
  (\glsenentrylong{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsenentryshort{##1}}##2\space
  (\glsenentrylong{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
  \protect\firstacronymfont{\glsenentryshortpl{##1}}##2\space
  (\glsenentrylongpl{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsenentryshortpl{##1}}##2\space
  (\glsenentrylongpl{##1})%
}%

\renewcommand*{\acronymentry}[1]{\acronymfont{\glsenentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}

```

long-sc-short $\langle long \rangle$ ($\backslash\text{textsc}\{\langle short \rangle\}$) acronym style.

```

\newacronymstyle{long-sc-short}%
{%
  \GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}

```

long-sm-short $\langle long \rangle$ ($\backslash\text{textsmaller}\{\langle short \rangle\}$) acronym style.

```

\newacronymstyle{long-sm-short}%
{%
  \GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glssacrpluralsuffix}%
}

```

sc-short-long $\langle short \rangle$ ($\backslash\text{textsc}\{\langle long \rangle\}$) 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` $\langle long \rangle$ (`\textsc{\langle short \rangle}`) acronym style that has an accompanying description (which the user needs to supply).

```
\newacronymstyle{long-sc-short-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{long-sc-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sc-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
```

`long-sm-short-desc` $\langle long \rangle$ (`\textsmaller{\langle short \rangle}`) acronym style that has an accompanying description (which the user needs to supply).

```
\newacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
```

`short-long-desc` $\langle short \rangle$ (`\{\langle long \rangle\}`) acronym style that has an accompanying description (which the user needs to supply).

```
\newacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
```

`sc-short-long-desc` $\langle long \rangle$ (`\textsc{\langle short \rangle}`) acronym style that has an accompanying description (which the user needs to supply).

```
\newacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
```

```

\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}%
{%
  \GlsUseAcrEntryDisplayStyle{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:

```

\glssupercase
{\glsentrylongpl{\glslabel}\glsinsert}%
}%

```

```

    }%
    {%
Singular form
    \glscapscase
    {%
Singular form, don't adjust case:
    \glentrylong{\glslabel}\glsinsert
    }%
    {%
Subsequent singular form, make first letter upper case:
    \Glsentrylong{\glslabel}\glsinsert
    }%
    {%
Subsequent singular form, all caps:
    \glssupercase
    {\glentrylong{\glslabel}\glsinsert}%
    }%
    }%
    }%
    {%
Not an acronym:
    \glsgenentryfmt
    }%
    }%
    {\glscustomtext\glsinsert}%
    }%
    {%
    \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%

    \renewcommand*{\acrfullfmt}[3]{%
        \glslink[##1]{##2}{\glentrylong{##2}##3\space
        (\acronymfont{\glentryshort{##2}})}}%
    \renewcommand*{\Acrfullfmt}[3]{%
        \glslink[##1]{##2}{\Glsentrylong{##2}##3\space
        (\acronymfont{\glentryshort{##2}})}}%
    \renewcommand*{\ACRfullfmt}[3]{%
        \glslink[##1]{##2}{%
        \glssupercase{\glentrylong{##2}##3\space
        (\acronymfont{\glentryshort{##2}})}}}%

    \renewcommand*{\acrfullplfmt}[3]{%
        \glslink[##1]{##2}{\glentrylongpl{##2}##3\space
        (\acronymfont{\glentryshortpl{##2}})}}%

    \renewcommand*{\Acrfullplfmt}[3]{%
        \glslink[##1]{##2}{\Glsentrylongpl{##2}##3\space
        (\acronymfont{\glentryshortpl{##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}{\glsacronymfont}{\glsentryfmt}%
}%
{%
  \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}{%
    \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]{%
    \textsc{\glsentrylong{##1}}\space (\acronymfont{\glsentryshort{##1}})}%
  }
}
```

```

        \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
    }

footnote-sm-desc \textsmaller{short}\footnote{long} acronym style that has an accom-
panying 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]{%
            \glentrylong{##1}\space (\acronymfont{\glentryshort{##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\aclong

Plural long form

\aclp
    \let\aclp\aclongpl

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

```

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

```

\newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
  \def\glsentryfmt[#1]{\glsentryfmt}%
}

```

The following commands for setting the acronym style are now deprecated.

`\DefaultNewAcronymDef` Sets up the acronym definition for the default style. The information is provided by the tokens `\glslabeltok`, `\glsshorttok`, `\glslongtok` and `\glskeylisttok`. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\noexpand@assign@default@plurals` `\@@gls@noexpand@assign@default@plurals` was used in `\DefaultNewAcronymDef`. Added in v4.47 to fix bug #162 (<https://www.dickimaw-books.com/bugtracker.php?key=162>). Removed in v4.50 along with the deprecated styles.

`\@@gls@@default@acr@plural`

`\gls@@default@acr@firstplural`

`\gls@@default@acr@descplural`

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

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

```

\newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}

```

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

```

\newcommand*{\acrlinkfootnote}[3]{%
  \footnote{\glslink{#1}{#2}{#3}}%
}

```

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

```

\newcommand*{\acrnoflinkfootnote}[3]{%
  \footnote{#3}%
}

```

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

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

<code>\descriptionFootnoteAcronymStyle</code>	If a description and footnote are both required, store the long form in the <code>symbol</code> key. Store the short form in <code>text</code> key. Note that since the long form is stored in the <code>symbol</code> key, if you want the long form to appear in the list of acronyms, you need to use a glossary style that displays the <code>symbol</code> key. This style was deprecated with the introduction of <code>\newacronymstyle</code> . Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\descriptionDUAAcronymDisplayStyle</code>	Sets the acronym display style for given glossary with description and dua combination. Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\DescriptionDUANewAcronymDef</code>	Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\descriptionDUAAcronymStyle</code>	Description, don't use acronym and no footnote. Note that the short form is stored in the <code>symbol</code> key, so if the short form needs to be displayed in the glossary, use a style that displays the <code>symbol</code> . This style was deprecated with the introduction of <code>\newacronymstyle</code> . Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\descriptionAcronymDisplayStyle</code>	Sets the acronym display style for given glossary using the description setting (but not footnote or dua). Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\DescriptionNewAcronymDef</code>	Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\SetDescriptionAcronymStyle</code>	Option description is used, but not dua or footnote. Store long form in first key and short form in <code>text</code> and <code>symbol</code> key. The name is stored using <code>\acronymformat</code> to allow the user to override the way the name is displayed in the list of acronyms. Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\FootnoteAcronymDisplayStyle</code>	Sets the acronym display style for given glossary with footnote setting (but not description or dua). Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\FootnoteNewAcronymDef</code>	Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\SetFootnoteAcronymStyle</code>	If footnote package option is specified, set the first use to append the long form (stored in <code>description</code>) as a footnote. Use the <code>description</code> key to store the long form. This style was deprecated with the introduction of <code>\newacronymstyle</code> . Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\glsdoparenifnotempty</code>	Do a space followed by the argument if the argument doesn't expand to empty or <code>\relax</code> . If argument isn't empty (or <code>\relax</code>), 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.

`\SetDUASStyle` 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@glstetoklabel{#1}@level\endcsname
}
```

\showglotext

\showglotext{<label>}

```
\newcommand*{\showglotext}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@text\endcsname
}
```

\showgloplural

\showgloplural{<label>}

```
\newcommand*{\showgloplural}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@plural\endcsname
}
```

\showglofirst

\showglofirst{<label>}

```
\newcommand*{\showglofirst}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@first\endcsname
}
```

\showglofirstpl

\showglofirstpl{<label>}

```
\newcommand*{\showglofirstpl}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@firstpl\endcsname
}
```

\showglotype

\showglotype{<label>}

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


\showglocounter

\showglocounter{<label>}

```
\newcommand*{\showglocounter}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@counter\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
}
```

\showglouseriiii

\showglouseriiii{<label>}

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

\showgloname

\showgloname{<label>}

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

\showglodesc

\showglodesc{<label>}

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

\showglodescplural

\showglodescplural{<label>}

```
\newcommand*{\showglodescplural}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@descplural\endcsname
}
```

\showglosort

\showglosort{<label>}

```
\newcommand*{\showglosort}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@sort\endcsname
}
```

\showglosymbol

\showglosymbol{<label>}

```
\newcommand*{\showglosymbol}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@symbol\endcsname
}
```

\showglosymbolplural

\showglosymbolplural{<label>}

```
\newcommand*{\showglosymbolplural}[1]{%  
  \expandafter\show\csname glo@glstdetoklabel{#1}@symbolplural\endcsname  
}
```

\showgloshort

\showgloshort{<label>}

```
\newcommand*{\showgloshort}[1]{%  
  \expandafter\show\csname glo@glstdetoklabel{#1}@short\endcsname  
}
```

\showglolong

\showglolong{<label>}

```
\newcommand*{\showglolong}[1]{%  
  \expandafter\show\csname glo@glstdetoklabel{#1}@long\endcsname  
}
```

\showgloindex

\showgloindex{<label>}

```
\newcommand*{\showgloindex}[1]{%  
  \expandafter\show\csname glo@glstdetoklabel{#1}@index\endcsname  
}
```

\showgloflag

\showgloflag{<label>}

```
\newcommand*{\showgloflag}[1]{%  
  \expandafter\show\csname ifglo@glstdetoklabel{#1}@flag\endcsname  
}
```

\showgloloclist

\showgloloclist{<label>}

```
\newcommand*{\showgloloclist}[1]{%  
  \expandafter\show\csname glo@glstdetoklabel{#1}@loclist\endcsname  
}
```

`\showglofield`

`\showglofield{<label>}{<field>}`

```
\newcommand*{\showglofield}[2]{%
\cshow{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}
\DeclareCurrentRelease{v4.51}{2022-11-02}
```

Package Definition:

\ProvidesPackage{glossary-hypernav}[2022/11/02 v4.51 (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}
```

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

\glsnavhypertarget

This command makes ⟨text⟩ a hypertarget for the glossary group whose label is given by ⟨label⟩ in the glossary given by ⟨type⟩. If ⟨type⟩ is omitted, \@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}%
}
```

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.

```
\@glstarget{\glsnavhyperlinkname{#1}{#2}}{#3}%
```

Check list of known groups to determine if a re-run is required.

```
\expandafter\let
\expandafter\@gls@list\csname @gls@hypergroup@#1\endcsname
```

Iterate through list and terminate loop if this group is found.

```
\@for\@gls@elem:=\@gls@list\do{%
\ifthenelse{\equal{\@gls@elem}{#2}}{\@endfortrue}{}}%
```

Check if list terminated prematurely.

```
\if@endfor
\else
```

This group was not included in the list, so issue a warning.

```
\GlossariesWarningNoLine{Navigation panel
for glossary type ‘#1’^^Jmissing group ‘#2’}%
\gdef\gls@hypergroup@rerun{%
\GlossariesWarningNoLine{Navigation panel
has changed. Rerun LaTeX}}%
\fi
}
```

`\gls@hypergroup@rerun` Give a warning at the end if re-run required

```
\let\gls@hypergroup@rerun\relax
\AtEndDocument{\gls@hypergroup@rerun}
```

`\@gls@hypergroup` This adds to (or creates) the command `\@gls@hypergroup@list@<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]{%
\ifcsdef{\@gls@hypergroup@list@#1}%
{\protected@csxappto{\@gls@hypergroup@list@#1}{, #2}}%
{\protected@csxdef{\@gls@hypergroup@list@#1}{#2}}%
}
```

The `\glsnavigation` command displays a simple glossary group navigation. The symbol and number elements are defined separately, so that they can be suppressed if need be. (In earlier versions this command will produce a link to all 28 groups, but some groups may not be defined if there are groups that do not contain any terms, in which case you will get an undefined hyperlink warning. Version 1.14 changed this to only use labels for groups that are present.) Now for the whole navigation bit:

`\glsnavigation`

```
\newcommand*{\glsnavigation}{%
\def\@gls@between{}%
\ifcsundef{\@gls@hypergroup@list@\currentglossary}%
```

```

{%
  \def\@gls@list{}%
}%
{%

  \letcs\@gls@list{\@gls@hypergroup\list@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
}%
}

```

`\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
}

```

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{v4.51}{2022-11-02}

```

Declare package:

```
\ProvidesPackage{glossary-inline}[2022/11/02 v4.51 (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%
    }%
  }%
  {%
    \ifglsdesc{##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{v4.51}{2022-11-02}
```

Declare package:

```
\ProvidesPackage{glossary-list}[2022/11/02 v4.51 (NLCT)]
```

`\indexspace` There are a few classes that don't define `\indexspace`, so provide a definition if it hasn't been defined.

```
\providecommand{\indexspace}{%  
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax  
}
```

`\glslistgroupheaderfmt` Provide a way of adjusting the format of the group headings.

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

`\glslistnavigationitem` Provide a way of adjusting the format of the navigation header. This puts the navigation line inside the optional argument of `item` to prevent unwanted space occurring at the start, but this can cause a problem if the navigation line is too long. With this command, it makes it easier for the user to customise the style without having to remember to modify `\glossaryheader` after the style has been set.

```
\newcommand*{\glslistnavigationitem}[1]{\item[#1]}
```

`\glslistinit` Provide a way to ensure appropriate expansion of the name if `entrycounter` is used with `getttitlestring`:

```
\newcommand{\glslistinit}{%  
  \ifdef\GetTitleStringDisableCommands  
  {%  
    \GetTitleStringSetup{expand}%  
    \GetTitleStringDisableCommands{%  
      \let\glsentryitem\@gobble  
      \let\glstarget\@secondoftwo  
      \let\glossentryname\glslistexpandedname
```

```

        \let\glslistgroupheaderfmt\@firstofone
        \let\glsgetgrouptitle\@firstofone
        \let\glsnavhypertarget\@secondoftwo
        \let\glsnavigation\relax
    }%
}%
}%
}%
}

```

`\glslistexpandedname`

```

\newcommand{\glslistexpandedname}[1]{%
\glsunexpandedfieldvalue{#1}{name}%
}

```

list The list glossary style uses the description environment. The group separator `\glsgroupskip` is redefined as `\indexspace` which produces a gap between groups. The glossary heading and the group headings do nothing. Sub-entries immediately follow the main entry without the sub-entry name. This style does not use the entry's symbol. This is used as the default style for the glossaries package.

```
\newglossarystyle{list}{%
```

Use description environment:

```

\renewenvironment{theglossary}%
{\glslistinit\begin{description}}{\end{description}}%

```

No header at the start of the environment:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries start a new item in the list:

```

\renewcommand*{\glossentry}[2]{%
\item[\glsentryitem{##1}%
\glstarget{##1}{\glossentryname{##1}}]
\glossentrydesc{##1}\glspostdescription\space ##2}%

```

Sub-entries continue on the same line:

```

\renewcommand*{\subglossentry}[3]{%
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\space
\glossentrydesc{##2}\glspostdescription\space ##3.}%

```

Add vertical space between groups:

```

\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}

```

listgroup The listgroup style is like the list style, but the glossary groups have headings.

```
\newglossarystyle{listgroup}{%
```

Base it on the list style:

```
\setglossarystyle{list}%
```

Each group has a heading:

```
\renewcommand*{\glsgroupheading}[1]{%  
  \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
```

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

```
\newglossarystyle{listhypergroup}{%
```

Base it on the list style:

```
\setglossarystyle{list}%
```

Add navigation links at the start of the environment.

```
\renewcommand*{\glossaryheader}{%  
  \glslistnavigationitem{\glsnavigation}}%
```

Each group has a heading with a hypertarget:

```
\renewcommand*{\glsgroupheading}[1]{%  
  \item[\glslistgroupheaderfmt  
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}]}
```

altlist The altlist glossary style is like the list style, but places the description on a new line. Sub-entries follow in separate paragraphs without the sub-entry name. This style does not use the entry's symbol.

```
\newglossarystyle{altlist}{%
```

Base it on the list style:

```
\setglossarystyle{list}%
```

Main (level 0) entries start a new item in the list with a line break after the entry name:

```
\renewcommand*{\glossentry}[2]{%  
  \item[\glsentryitem{##1}%  
    \glstarget{##1}{\glossentryname{##1}}]}
```

Version 3.04 changed `\newline` to the following paragraph break stuff (thanks to Daniel Gebhardt for supplying the fix) to prevent a page break occurring at this point.

```
\mbox{} \par \nobreak \@afterheading  
\glossentrydesc{##1} \glspostdescription \space ##2}%
```

Sub-entries start a new paragraph:

```
\renewcommand{\subglossentry}[3]{%  
  \par  
  \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{\glsgrouptitle{##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{\glsgroupnavigation}}%
```

Each group has a heading with a hypertext:

```
\renewcommand*{\glsgroupheading}[1]{%
  \item[\glslistgroupheaderfmt
    {\glsgrouphypertarget{##1}{\glsgrouptitle{##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]{%
    \glssentryitem{##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[\glstentryitem{##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{v4.51}{2022-11-02}
```

Declare package:

```
\ProvidesPackage{glossary-long}[2022/11/02 v4.51 (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]{%
  \glstryitem{##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{\glstdescwidth}|}{\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{\glsgdescwidth}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]{%
\glstentryitem{##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>)

```
\ifglsgnogroupskip
\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 glosstylelong3col style:

```
\setglossarystyle{long3col}%
```

Use a longtable with 3 columns with vertical lines around them:

```
\renewenvironment{theglossary}%  
{\begin{longtable}{|l|p{\glsgdescwidth}|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 glosstylelong3col 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 glosstylelong3colborder 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{\glstdescwidth}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{\glstdescwidth}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{v4.51}{2022-11-02}
```

Declare package:

```
\ProvidesPackage{glossary-longbooktabs}[2022/11/02 v4.51 (NLCT)]
```

Requires `booktabs` package:

```
\RequirePackage{booktabs}
```

and the base packages for long styles:

```
\RequirePackage{glossary-long}  
\RequirePackage{glossary-longragged}
```

(`longtable` and `array` loaded by those packages).

long-booktabs The `long-booktabs` style is similar to the `longheader` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{long-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

As with the `longheader` style, use the `long` style as a base.

```
\setglossarystyle{long}%
```

Add a header with rules.

```
\renewcommand*{\glossaryheader}{%
\toprule \bfseries \entryname & \bfseries
\descriptionname\tabularnewline\midrule\endhead
\bottomrule\endfoot}%
```

Check for the `nogroupskip` package option. If there should be a gap between groups, insert the penalty and the vertical space. The check for `nogroupskip` should occur outside `\glsgroupskip` to be on the safe side.

```
\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}
```

long3col-booktabs The `long3col-booktabs` style is similar to the `long3colheader` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{long3col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the `long3col` style as a base.

```
\setglossarystyle{long3col}%
```

Add a header with rules.

```
\renewcommand*{\glossaryheader}{%
\toprule \bfseries \entryname &
\bfseries \descriptionname &
\bfseries \pagelistname
\tabularnewline\midrule\endhead
\bottomrule\endfoot}%
```

Check for the `nogroupskip` package option. If there should be a gap between groups, insert the penalty and the vertical space. The check for `nogroupskip` should occur outside `\glsgroupskip` to be on the safe side.

```
\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
```

```

\else
\renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}

```

long4col-booktabs The long4col-booktabs style is similar to the long4colheader style but uses the booktabs rules and patches longtable to check for group skip occurring at a page break.

```
\newglossarystyle{long4col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the long4col style as a base.

```
\setglossarystyle{long4col}%
```

Add a header with rules.

```

\renewcommand*{\glossaryheader}{%
\toprule \bfseries \entryname &
\bfseries \descriptionname &
\bfseries \symbolname &
\bfseries \pagelistname
\tabularnewline\midrule\endhead
\bottomrule\endfoot}%

```

Check for the nogroupskip package option. If there should be a gap between groups, insert the penalty and the vertical space. The check for nogroupskip should occur outside \glsgroupskip to be on the safe side.

```

\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}

```

altlong4col-booktabs The altlong4col-booktabs style is similar to the altlong4colheader style but uses the booktabs rules and patches longtable to check for group skip occurring at a page break.

```
\newglossarystyle{altlong4col-booktabs}{%
```

The patch \glspatchLToutput is already applied in long4col-booktabs and so doesn't need to be here.

```
\glspatchLToutput
```

Use the long4col-booktabs style as a base.

```
\setglossarystyle{long4col-booktabs}%
```

Change the column specifications:

```

\renewenvironment{theglossary}%
{\begin{longtable}{lp{\glstdescwidth}lp{\glspagelistwidth}}}%
{\end{longtable}}%
}

```

Ragged styles.

longragged-booktabs The `longragged-booktabs` style is similar to the `longragged` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{longragged-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the `long-booktabs` style as a base.

```
\setglossarystyle{long-booktabs}%
```

Adjust the column specification.

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}}}%  
  {\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{\glsgdescwidth}%  
    >{\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\vsizel\@colht
{\unvbox\z@\box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%
\fi
\else
\setbox\@cclv\vbox{\unvbox\@cclv\copy\LT@foot\vss}%
\@makecol
\@outputpage
}
```

```

\global\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{v4.51}{2022-11-02}

```

Declare package:

```
\ProvidesPackage{glossary-longragged}[2022/11/02 v4.51 (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}[1>{\raggedright}p{\glsdescwidth}%  
    >{\raggedright}p{\glspagelistwidth}}}%  
  {\end{longtable}}%
```

No table header:

```
\renewcommand*{\glossaryheader}{}%
```

No headings between groups:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
\renewcommand{\glossentry}[2]{%  
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &  
  \glossentrydesc{##1} & ##2\tabularnewline  
}%
```

Sub-entries on a separate row (no name, description in second column, page list in third column):

```
\renewcommand{\subglossentry}[3]{%  
  &  
  \glssubentryitem{##2}%  
  \glstarget{##2}{\strut}\glossentrydesc{##2} &  
  ##3\tabularnewline  
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
\fi
}
```

`longragged3colborder` The `longragged3colborder` style is like the `longragged3col` style but with a border:

```
\newglossarystyle{longragged3colborder}{%
```

Base it on the `glostylelongragged3col` style:

```
\setglossarystyle{longragged3col}%
```

Use a `longtable` with 3 columns with vertical lines around them:

```
\renewenvironment{theglossary}%
{\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|%
>{\raggedright}p{\glspagelistwidth}|}%
{\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
```

`longragged3colheader` The `longragged3colheader` style is like `longragged3col` but with a header row:

```
\newglossarystyle{longragged3colheader}{%
```

Base it on the `glostylelongragged3col` style:

```
\setglossarystyle{longragged3col}%
```

Set the table's header:

```
\renewcommand*{\glossaryheader}{%
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries\pagelistname\tabularnewline\endhead}%
}
```

`longragged3colheaderborder` The `longragged3colheaderborder` style is like the above but with a border

```
\newglossarystyle{longragged3colheaderborder}{%
```

Base it on the `glostylelongragged3colborder` style:

```
\setglossarystyle{longragged3colborder}%
```

Set the table's header and add horizontal line at table's foot:

```
\renewcommand*{\glossaryheader}{%
  \hline
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries\pagelistname\tabularnewline\hline\endhead
  \hline\endfoot}%
}
```

`altlongragged4col` The `altlongragged4col` style is like the `altlong4col` style defined in the `glossary-long` package, except that ragged right formatting is used for the description and page list columns.

```
\newglossarystyle{altlongragged4col}{%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
{\begin{longtable}[1>{\raggedright}p{\glsdescwidth}1%
>{\raggedright}p{\glspagelistwidth}}}%
{\end{longtable}}%
```

No table header:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a single row (name in first column, description in second column, symbol in third column, page list in last column):

```
\renewcommand{\glossentry}[2]{%
\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1} & \glossentrysymbol{##1} &
##2\tabularnewline
}%
```

Sub entries on a single row with no name (description in second column, symbol in third column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2} &
\glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
\fi
}
```

`altlongragged4colheader` The `altlongragged4colheader` style is like `altlongragged4col` but with a header row.

```
\newglossarystyle{altlongragged4colheader}{%
```

Base it on the `glostylealtlongragged4col` style:

```
\setglossarystyle{altlongragged4col}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{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}%
}
```

`altlongragged4colheaderborder` 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{v4.51}{2022-11-02}
```

Declare package:

```
\ProvidesPackage{glossary-mcols}[2022/11/02 v4.51 (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{\glsgrouptitle{##1}}\indexspace}%
}
```


mcolindexhypergroup The mcolindexhypergroup style is like the mcolindexgroup style but has hyper navigation.

```
\newglossarystyle{mcolindexhypergroup}{%
```

Base it on the glostylemcolindex style:

```
\setglossarystyle{mcolindex}{%
```

Put navigation links to the groups at the start of the glossary:

```
\renewcommand*{\glossaryheader}{%
\item\glstreenavigationfmt{\glsnavigation}\indexspace}%
```

Add a heading for each group (with a target). The group's title is in bold followed by a vertical gap.

```
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
\indexspace}%
}
```

mcolindexspannav Similar to mcolindexhypergroup, but puts the navigation line in the optional argument of multicols.

```
\newglossarystyle{mcolindexspannav}{%
```

```
\setglossarystyle{index}{%
```

```
\renewenvironment{theglossary}{%
```

```
{%
```

```
\begin{multicols}{\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{\glsgrouptitle{##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}{\glsgrouptitle{##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}{\glsgrouptitle{##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{\glsgrouptitle{##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}{\glsgrouptitle{##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{\glsnavigationtarget{##1}{\glsgrouptitle{##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{\glsgrouptitle{##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{\glssnavhypertarget{##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{\glssnavigation}]
    \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{\glssnavhypertarget{##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{v4.51}{2022-11-02}
```

Declare package:

```
\ProvidesPackage{glossary-super}[2022/11/02 v4.51 (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]{%
  \glssentryitem{##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{\glstdescwidth}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]{%  
  \glssentryitem{##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{\glstdescwidth}|p{\glspagelistwidth}|}%
        {\end{supertabular}}%
    }

```

super3colheader The `super3colheader` style is like the `super3col` style but with a header row:

```

    \newglossarystyle{super3colheader}{%
Base it on the glostylessuper3col style:
    \setglossarystyle{super3col}%
Put the glossary in a supertabular environment with three columns, a header
and no tail:
    \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    \begin{supertabular}{|p{\glstdescwidth}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 glostylessuper3colborder 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{\glstdescwidth}|p{\glspagelistwidth}|}%
    {\end{supertabular}}%
    }

```

super4col The `super4col` glossary style has four columns, where the third column contains the value of the corresponding symbol key used when that entry was defined.

```

    \newglossarystyle{super4col}{%
Put the glossary in a supertabular environment with four columns and no head
or tail:
    \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}}%
    \begin{supertabular}{|l|l|l|l|}%
    \end{supertabular}%
Do nothing at the start of the table:
    \renewcommand*{\glossaryheader}{}%

```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row with the name in the first column, description in second column, symbol in third column and page list in last column:

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  \glossentrysymbol{##1} & ##2\tabularnewline
}%
```

Sub entries on a row with no name, the description in the second column, symbol in third column and page list in last column:

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  \glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for nogroupskip must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi
}
```

super4colheader The `super4colheader` style is like the `super4col` but with a header row.

```
\newglossarystyle{super4colheader}{%
```

Base it on the `glostylesuper4col` style:

```
\setglossarystyle{super4col}%
```

Put the glossary in a `supertabular` environment with four columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
    \bfseries\symbolname &
    \bfseries\pagelistname\tabularnewline}%
  \tabletail{}}%
  \begin{supertabular}{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}{|1|1|1|1|}%
  {\end{supertabular}}%
```

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

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

Base it on the `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}{|1|1|1|1|}%
  {\end{supertabular}}%
```

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

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

Base it on the `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{\glsgdescwidth}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}%
    {\lllp{\glsgdescwidth}\lllp{\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}%
    {\lllp{\glsgdescwidth}\lllp{\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{v4.51}{2022-11-02}
```

Declare package:

```
\ProvidesPackage{glossary-superragged}[2022/11/02 v4.51 (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]{%  
  \glstryitem{##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{\glsgdescwidth}}}%
  {\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{\glsgdescwidth}%
    >{\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]{%
  \glsglentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  ##2\tabularnewline
}%
```

Sub entries on a row (no name, description in second column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& \tabularnewline}%
\fi
}
```

superragged3colborder The `superragged3colborder` style is like the `superragged3col` style, but with a border:

```
\newglossarystyle{superragged3colborder}{%
```

Base it on the `glostylesuperragged3col` style:

```
\setglossarystyle{superragged3col}%
```

Put the glossary in a `supertabular` environment with three columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
    >{\raggedright}p{\glspagelistwidth}|}%
  {\end{supertabular}}%
}
```

superragged3colheader The `superragged3colheader` style is like the `superragged3col` style but with a header row:

```
\newglossarystyle{superragged3colheader}{%
```

Base it on the `glostylesuperragged3col` style:

```
\setglossarystyle{superragged3col}%
```

Put the glossary in a `supertabular` environment with three columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline}\tabletail{}%
  \begin{supertabular}{|l>{\raggedright}p{\glsdescwidth}%
    >{\raggedright}p{\glspagelistwidth}}%
  {\end{supertabular}}%
}
```

raggedright3colheaderborder The `superragged3colheaderborder` style is like the `superragged3col` style but with a header and border:

```
\newglossarystyle{superragged3colheaderborder}{%
```

Base it on the `glostylesuperragged3colborder` style:

```
\setglossarystyle{superragged3colborder}%
```


Put the glossary in a `supertabular` environment with three columns, a header with horizontal lines and a horizontal line in the tail:

```
\renewenvironment{theglossary}%
{
  \tablehead{\hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline}%
  \tabletail{\hline}%
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
    >{\raggedright}p{\glspagelistwidth}|}%
  \end{supertabular}}%
}
```

`altsuperragged4col` The `altsuperragged4col` glossary style is like `altsuper4col` style in the `glossary-super` package but uses ragged right formatting in the description and page list columns.

```
\newglossarystyle{altsuperragged4col}{%
```

Put the glossary in a `supertabular` environment with four columns and no head or tail:

```
\renewenvironment{theglossary}%
{
  \tablehead{} \tabletail{}%
  \begin{supertabular}{|l>{\raggedright}p{\glsdescwidth}l%
    >{\raggedright}p{\glspagelistwidth}|}%
  \end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row with the name in the first column, description in second column, symbol in third column and page list in last column:

```
\renewcommand{\glossentry}[2]{%
  \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  \glossentrysymbol{##1} & ##2\tabularnewline
}%
```

Sub entries on a row with no name, the description in the second column, symbol in third column and page list in last column:

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  \glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```

\ifglsgroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{& & & \tabularnewline}%
\fi
}

```

altsuperragged4colheader The `altsuperragged4colheader` style is like the `altsuperragged4col` style but with a header row.

```
\newglossarystyle{altsuperragged4colheader}{%
```

Base it on the `glostylealtsuperragged4col` style:

```
\setglossarystyle{altsuperragged4col}%
```

Put the glossary in a `supertabular` environment with four columns, a header and no tail:

```

\renewenvironment{theglossary}%
{
\tablehead{\bfseries\entryname&\bfseries\descriptionname&
\bfseries\symbolname &
\bfseries\pagelistname\supertabularnewline}\tabletail{}%
\begin{supertabular}{1>\raggedright}p{\glsgdescwidth}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{\glsgdescwidth}|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}\l|}%
  {\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{v4.51}{2022-11-02}
```

Declare package:

```
\ProvidesPackage{glossary-tree}[2022/11/02 v4.51 (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 `\glstnamefont`.) This command was previously also used to format the group headings.

```
\newcommand*{\glstreenamefmt}[1]{\textbf{#1}}
```

`\glstreegroupheaderfmt` Format used to display the group header in the tree styles. Before v4.22, `\glstreenamefmt` was used for the group header, so the default definition uses that to help maintain backward-compatibility, since in previous versions redefining `\glstreenamefmt` would've also affected the group headings.

```
\newcommand*{\glstreegroupheaderfmt}[1]{\glstreenamefmt{#1}}
```

`\glstreenavigationfmt` Format used to display the navigation header in the tree styles.

```
\newcommand*{\glstreenavigationfmt}[1]{\glstreenamefmt{#1}}
```

Allow the user to adjust the index style without disturbing the index.

`\glstreeitem` Top level item used in index style.

```
\ifdef\@idxitem
{\newcommand{\glstreeitem}{\@idxitem}}
{\newcommand{\glstreeitem}{\par\hangindent40\p@}}
```

`\glstreesubitem` Level 1 item used in index style.

```
\ifdef\subitem
{\let\glstreesubitem\subitem}
{\newcommand\glstreesubitem{\glstreeitem\hspace*{20\p@}}}
```

`\glstreesubsubitem` Level 1 item used in index style.

```
\ifdef\subsubitem
{\let\glstreesubsubitem\subsubitem}
{\newcommand\glstreesubsubitem{\glstreeitem\hspace*{30\p@}}}
```

`\glstreepredesc` Allow the user to adjust the space before the description (except for the `almtree` style).

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

`\glstreechildpredesc` Allow the user to adjust the space before the description for sub-entries (except for the `treenoname` and `almtree` style).

```
\newcommand{\glstreechildpredesc}{\space}
```

`index` The index glossary style is similar in style to the way indices are usually typeset using `\item`, `\subitem` and `\subsubitem`. The entry name is set in bold. If an entry has a symbol, it is placed in brackets after the name. Then the description is displayed, followed by the number list. This style allows up to three levels.

```
\newglossarystyle{index}{%
```

Set the paragraph indentation and skip and define `\item` to be the same as that used by `theindex`:

```
\renewenvironment{theglossary}%
{\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
\let\item\glstreeitem
\let\subitem\glstreesubitem
\let\subsubitem\glstreesubsubitem
}%
{\par}%
```

Do nothing at the start of the environment:

```
\renewcommand*{\glossaryheader}{}%
```

No group headers:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entry starts a new item with the name in bold followed by the symbol in brackets (if it exists), the description and the page list.

```
\renewcommand*{\glossentry}[2]{%
  \item\glstentryitem{##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}{\ifglsgnpgroupskip\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}{\glsgrouptitle{##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
  \glstryitem{##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}{\ifglsgnigroupskip\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{\glsgrouptitle{##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
{\glsnavigationhypertarget{##1}{\glsgrouptitle{##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*{\glsgrouphheading}[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
  \glstryitem{##1}\glstreenamfmt{\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{\glsgrouphheading}[1]{\par
  \noindent\glstreegroupheaderfmt
  {\glsggetgrouptitle{##1}}\par\indexspace}%
}
```

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

```
\newglossarystyle{treenamehypergroup}{%
```

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
\glstreenameheaderfmt
{\glsnavigationtarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}
```

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

```
\newrobustcmd*{\glswidesttoplevelname}[1][\glo@types]{%
\dimen@=0pt\relax
\gls@tmplen=0pt\relax
\forall glossaries[#1]{\gls@type}%
{%
\forall glsentries[\gls@type]{\glo@label}%
{%
\ifglshasparent{\glo@label}%
}%
{%
\glsmeasurewidth{\dimen@}%
{\glstreenamefmt{\glsentryname{\glo@label}}}%
\ifdim\dimen@>\gls@tmplen
\gls@tmplen=\dimen@
\letcs{\glswidestname}{glo@\glsdetoklabel{\glo@label}@name}%
\fi
}%
}%
}%
}
```

`\glsssetwidest` `\glsssetwidest[level]{text}` sets the widest text for the given level. It is used by the `alttree` glossary styles to determine the indentation of each level.

```
\newcommand*{\glsssetwidest}[2][0]{%
\expandafter\def\csname @glswidestname\romannumeral#1\endcsname{%
#2}%
}
```

`\@glswidestname` Initialise `\@glswidestname`.

```
\newcommand*{\@glswidestname}{}
```

`\glstreenamebox` Used by the `alttree` style to create the box for the name and associated information.

```
\newcommand*{\glstreenamebox}[2]{%
\makebox[#1][l]{#2}%
}
```

`alttree` The `alttree` glossary style is similar in style to the `tree` style, but the indentation is obtained from the width of `\@glswidestname` which is set using `\glsssetwidest`.

```
\newglossarystyle{alttree}{%
```

Redefine theglossary environment.

```
\renewenvironment{theglossary}%  
  {\def\@gls@prevlevel{-1}%  
   \mbox{}\par}%  
  {\par}%
```

Set the header and group headers to nothing.

```
\renewcommand*{\glossaryheader}{}%  
\renewcommand*{\glsgroupheading}[1]{}%
```

Redefine the way that the level 0 entries are displayed.

```
\renewcommand{\glossentry}[2]{%  
  \ifnum\@gls@prevlevel=0\relax  
  \else
```

Find out how big the indentation should be by measuring the widest entry.

```
    \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%  
  \fi
```

Set the hangindent and paragraph indent.

```
\hangindent\glstreeindent  
\parindent\glstreeindent
```

Put the name to the left of the paragraph block.

```
\makebox[0pt][r]{\glstreenamebox{\glstreeindent}{%  
  \glstryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}}}%
```

If the symbol is missing, ignore it, otherwise put it in brackets.

```
\ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}%
```

Do the description followed by the description terminator and location list.

```
\glossentrydesc{##1}\glspostdescription \space ##2\par
```

Set the previous level to 0.

```
\def\@gls@prevlevel{0}%  
}%
```

Redefine the way sub-entries are displayed.

```
\renewcommand{\subglossentry}[3]{%
```

Increment and display the sub-entry counter if this is a level 1 entry and the sub-entry counter is in use.

```
  \ifnum##1=1\relax  
    \glssubentryitem{##2}%  
  \fi
```

If the level hasn't changed, keep the same settings, otherwise adjust \glstreeindent accordingly.

```
  \ifnum\@gls@prevlevel=##1\relax  
  \else
```

Compute the widest entry for this level, or for level 0 if not defined for this level.

Store in `\gls@tmplen`

```
\ifundefined{@glswidestname\romannumeral##1}{%
  \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}{%
    \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{
      \csname @glswidestname\romannumeral##1\endcsname\space}}}%
```

Determine if going up or down a level

```
\ifnum\@gls@prevlevel<##1\relax
```

Depth has increased, so add the width of the widest entry to `\glstreeindent`.

```
\setlength\glstreeindent\gls@tmplen
\addtolength\glstreeindent\parindent
\parindent\glstreeindent
\else
```

Depth has decreased, so subtract width of the widest entry from the previous level to `\glstreeindent`. First determine the width of the widest entry for the previous level and store in `\glstreeindent`.

```
\ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
  \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{
    \@glswidestname\space}}{%
    \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{
      \csname @glswidestname\romannumeral\@gls@prevlevel
        \endcsname\space}}}%
```

Subtract this length from the previous level's paragraph indent and set to `\glstreeindent`.

```
\addtolength\parindent{-\glstreeindent}%
\setlength\glstreeindent\parindent
\fi
\fi
```

Set the hanging indentation.

```
\hangindent\glstreeindent
```

Put the name to the left of the paragraph block

```
\makebox[Opt][r]{\glstreenamebox{\gls@tmplen}{%
  \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
```

If the symbol is missing, ignore it, otherwise put it in brackets.

```
\ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
```

Do the description followed by the description terminator and location list.

```
\glossentrydesc{##2}\glspostdescription\space ##3\par
```

Set the previous level macro to the current level.

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

Vertical gap between groups is the same as that used by indices:

```
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
```

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

```
\newglossarystyle{alttreegroup}{%  
Base it on the glosstylealttree style:  
  \setglossarystyle{alttree}%  
Give each group a heading.  
  \renewcommand{\glsgroupheading}[1]{\par  
    \def\@gls@prevlevel{-1}%  
    \hangindent0pt\relax  
    \parindent0pt\relax  
    \glstreegroupheaderfmt{\glsgrouptitle{##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 glosstylealttree 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}{\glsgrouptitle{##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{v4.51}{2022-11-02}
```

Declare package

```
\ProvidesPackage{glossaries-babel}[2022/11/02 v4.51 (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{v4.51}{2022-11-02}
```

Declare package

```
\ProvidesPackage{glossaries-polyglossia}[2022/11/02 v4.51 (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{v4.51}{2022-11-02}
```

Declare package:

```
\ProvidesPackage{glossaries-prefix}[2022/11/02 v4.51 (NLCT)]
```

Pass all options to glossaries:

```
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
```

Process options:

```
\ProcessOptions
```

Load glossaries:

```
\RequirePackage{glossaries}
```

Add the new keys:

```
\define@key{glossentry}{prefixfirst}{\def\@glo@entryprefixfirst{#1}}%
\define@key{glossentry}{prefixfirstplural}{\def\@glo@entryprefixfirstplural{#1}}%
\define@key{glossentry}{prefix}{\def\@glo@entryprefix{#1}}%
\define@key{glossentry}{prefixplural}{\def\@glo@entryprefixplural{#1}}%
```

Add them to \@gls@keymap:

```
\appto\@gls@keymap{,%
  {prefixfirst}{prefixfirst},%
  {prefixfirstplural}{prefixfirstplural},%
  {prefix}{prefix},%
  {prefixplural}{prefixplural}}%
}
```

Set the default values:

```
\appto\@newglossaryentryprehook{%
  \def\@glo@entryprefix{}%
  \def\@glo@entryprefixplural{}%
  \let\@glo@entryprefixfirst\@gls@default@value
  \let\@glo@entryprefixfirstplural\@gls@default@value
}
```

Set the assignment code:

```
\appto\@newglossaryentryposthook{%
  \gls@assign@field{}\@glo@label}{prefix}\@glo@entryprefix}%
  \gls@assign@field{}\@glo@label}{prefixplural}\@glo@entryprefixplural}%
}
```

If prefixfirst has not been supplied, make it the same as prefix.

```
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @prefix\endcsname}\@glo@label}{prefixfirst}%
  {\@glo@entryprefixfirst}%
}
```

If prefixfirstplural has not been supplied, make it the same as prefixplural.

```
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @prefixplural\endcsname}\@glo@label}%
  {prefixfirstplural}\@glo@entryprefixfirstplural}%
}
```

Define commands to access these fields:

```
\glsentryprefixfirst
\newcommand*{\glsentryprefixfirst}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixfirst}}

\glsentryprefixfirstplural
\newcommand*{\glsentryprefixfirstplural}[1]{%
  \csuse{glo@\glsdetoklabel{#1}@prefixfirstplural}}

\glsentryprefix
\newcommand*{\glsentryprefix}[1]{\csuse{glo@\glsdetoklabel{#1}@prefix}}

\glsentryprefixplural
\newcommand*{\glsentryprefixplural}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixplural}}
```

Now for the initial upper case variants:

```
\Glsentryprefixfirst
\newcommand*{\Glsentryprefixfirst}[1]{%
  \@Glsentryfield{#1}{prefixfirst}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentryprefixfirst}{\Glentryprefixfirst}
```

\Glentryprefixfirstplural

```
\newcommand*{\Glentryprefixfirstplural}[1]{%
  \@Glentryfield{#1}{prefixfirstplural}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentryprefixfirstplural}{\Glentryprefixfirstplural}
```

\Glentryprefix

```
\newcommand*{\Glentryprefix}[1]{%
  \@Glentryfield{#1}{prefix}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentryprefix}{\Glentryprefix}
```

\Glentryprefixplural

```
\newcommand*{\Glentryprefixplural}[1]{%
  \@Glentryfield{#1}{prefixplural}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentryprefixplural}{\Glentryprefixplural}
```

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

\ifglshasprefix

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

\ifglshasprefixplural

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

\ifglshasprefixfirst

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


`\ifglshasprefixfirstplural`

```
\newcommand*{\ifglshasprefixfirstplural}[3]{%
  \ifcempty{glo@glsdetoklabel{#1}@prefixfirstplural}%
    {#3}%
    {#2}%
}
```

`\@glsprefix@record@hook` Need to take into account the possibility that glossaries-extra might be loaded with the `record` option.

```
\providecommand{\@glsprefix@record@hook}[2]{%
  \ifdef\@glsextr@record
    {\@glsextr@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}{\@glshyp@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}%
  \glstoifexists{#2}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefix{#2}{\glstryprefix{#2}\glsprefixsep}{}%
    }%
    {%
      \ifglshasprefixfirst{#2}{\glstryprefixfirst{#2}\glsprefixsep}{}%
    }%
    \@glso{#1}{#2}[#3]%
  }%
}
```

Similarly for the plural version:

```
\pglsp1
\newrobustcmd{\pglsp1}{\@gls@hyp@opt\@pglsp1}
```

\@pglsp1 Unstarred version.

```
\newcommand*{\@pglsp1}[2][\%
\new@ifnextchar[%
{\@pglsp1@{#1}{#2}}%
{\@pglsp1@{#1}{#2}[]}%
}
```

\@pglsp1@ Read in the final optional argument:

```
\def\@pglsp1@#1#2[#3]{%
\@glsprefix@record@hook{#1}{#2}%
\glsdoifexists{#2}%
{%
\ifglsused{#2}%
{%
\ifgls@hasprefixplural{#2}{\glsentryprefixplural{#2}\glsprefixsep}{}%
}%
{%
\ifgls@hasprefixfirstplural{#2}%
{\glsentryprefixfirstplural{#2}\glsprefixsep}{}%
}%
\@glspl@{#1}{#2}[#3]%
}%
}
```

Now for the first letter upper case versions:

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

Add sentence case mapping.

```
\glsmfuaddmap{\pgls}{\Pgls}
```

\@Pgls Unstarred version.

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

\@Pgls@ Read in the final optional argument:

```
\def\@Pgls@#1#2[#3]{%
\@glsprefix@record@hook{#1}{#2}%
\glsdoifexists{#2}%
{%
\ifglsused{#2}%
}
```

```

{%
  \ifglshasprefix{#2}%
  {%
    \Glentryprefix{#2}%
    \glsprefixsep
    \@gls@{#1}{#2}[#3]%
  }%
  {\@Gls@{#1}{#2}[#3]}%
}%
{%
  \ifglshasprefixfirst{#2}%
  {%
    \Glentryprefixfirst{#2}%
    \glsprefixsep
    \@gls@{#1}{#2}[#3]%
  }%
  {\@Gls@{#1}{#2}[#3]}%
}%
}%
}

```

Similarly for the plural version:

```

\Pglsp1
\newrobustcmd{\Pglsp1}{\@gls@hyp@opt\Pglsp1}

```

Add sentence case mapping.

```

\glsmfuaddmap{\pglsp1}{\Pglsp1}

```

\@Pglsp1 Unstarred version.

```

\newcommand*{\@Pglsp1}[2][{}]{%
  \new@ifnextchar[%
    {\@Pglsp1@{#1}{#2}}%
    {\@Pglsp1@{#1}{#2}[]}%
  }

```

\@Pglsp1@ Read in the final optional argument:

```

\def\@Pglsp1@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglshasprefixplural{#2}%
      {%
        \Glentryprefixplural{#2}%
        \glsprefixsep
        \@glspl@{#1}{#2}[#3]%
      }%
    }%
  }%
}

```

```

        {\@Glspl@{#1}{#2}[#3]}%
    }%
    {%
        \ifglshasprefixfirstplural{#2}%
        {%
            \Glsentryprefixfirstplural{#2}%
            \glsprefixsep
            \@Glspl@{#1}{#2}[#3]%
        }%
        {\@Glspl@{#1}{#2}[#3]}%
    }%
}
}

```

Finally the all upper case versions:

```

\PGLS
\newrobustcmd{\PGLS}{\@gls@hyp@opt\@PGLS}

```

\@PGLS Unstarred version.

```

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

```

\@PGLS@ Read in the final optional argument:

```

\def\@PGLS@#1#2[#3]{%
    \@glsprefix@record@hook{#1}{#2}%
    \glsdoifexists{#2}%
    {%
        \ifglsused{#2}%
        {%
            \ifglshasprefix{#2}%
            {\glsuppercase{\glsentryprefix{#2}\glsprefixsep}}{}%
        }%
        {%
            \ifglshasprefixfirst{#2}%
            {\glsuppercase{\glsentryprefixfirst{#2}\glsprefixsep}}{}%
        }%
        \@GLS@{#1}{#2}[#3]%
    }%
}

```

Plural version:

```

\PGLSpl
\newrobustcmd{\PGLSpl}{\@gls@hyp@opt\@PGLSpl}

```

\@PGLSp1 Unstarred version.

```
\newcommand*{\@PGLSp1}[2][\%
  \new@ifnextchar[\%
    {\@PGLSp1@{#1}{#2}}\%
    {\@PGLSp1@{#1}{#2}[]}\%
  }
```

\@PGLSp1@ Read in the final optional argument:

```
\def\@PGLSp1@#1#2[#3]{\%
  \@glsprefix@record@hook{#1}{#2}\%
  \glsdoifexists{#2}\%
  {\%
    \ifglsused{#2}\%
    {\%
      \ifglsahasprefixplural{#2}\%
      {\glsuppercase{\glsentryprefixplural{#2}\glsprefixsep}}{\}%
    }\%
    {\%
      \ifglsahasprefixfirstplural{#2}\%
      {\glsuppercase{\glsentryprefixfirstplural{#2}\glsprefixsep}}{\}%
    }\%
    \@GLSp1@{#1}{#2}[#3]\%
  }\%
}
```

6 Accessibility Support (glossaries-accsupp Code)

The glossaries-accsupp package is experimental. It is intended to provide a means of using the PDF accessibility support in glossary entries. See the accsupp documentation for further details about accessibility support.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossaries-accsupp-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-accsupp-2021-11-01.sty}
\DeclareCurrentRelease{v4.51}{2022-11-02}
```

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

```
\ProvidesPackage{glossaries-accsupp}[2022/11/02 v4.51 (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, `\@glxtr@doaccsupp` will have been set, otherwise it will be empty.

```
\ifx\@glxtr@doaccsupp\empty
\GlossariesWarning{The ‘glossaries-accsupp’
package has been loaded\MessageBreak
after the ‘glossaries-extra’ package. This\MessageBreak
can cause a failure to integrate both packages. \MessageBreak
Either use the ‘accsupp’ option when you load\MessageBreak
‘glossaries-extra’ or load ‘glossaries-accsupp’\MessageBreak
before loading ‘glossaries-extra’}%
\fi
}
{}
```

`\compatibleglossentry` Override style compatibility macros:

`\compatiblesubglossentry`

Required packages:

```
\RequirePackage{glossaries}
```

`\gls@accsupp@engine` There’s currently only support for `accsupp`, but if you define `\gls@accsupp@engine` before loading `glossaries-accsupp`, you can prevent `accsupp` from being loaded. Redefining this command after `glossaries-accsupp` has loaded obviously won’t do anything (and so is an internal command to deter casual use). If it is defined to something other than `accsupp` then `\gls@accessibility` will need to be defined to something appropriate.

```
\providecommand{\gls@accsupp@engine}{accsupp}
```

```
\gls@accessibility[\options]{\PDF element}{\value}{\content}
```

`\gls@accessibility`

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

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

`\glsaccessibility`

User-level command that includes debug info if required.

```
\newcommand{\glsaccessibility}[4][]{%  
  \@glsshowacccsupp{#1}{#2}{#3}%  
  \gls@accessibility{#1}{#2}{#3}{#4}%  
}
```

6.1 Defining Replacement Text

The version 0.1 stored the replacement text in the symbol key. This has been changed to use the new keys defined here. Example of use:

```
%\newglossaryentry{dr}{name=Dr,description={},access={Doctor}}  
%
```

access The replacement text corresponding to the name key:

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

textaccess The replacement text corresponding to the text key:

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

firstaccess The replacement text corresponding to the first key:

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

pluralaccess The replacement text corresponding to the plural key:

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

firstpluralaccess The replacement text corresponding to the firstplural key:

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

symbolaccess The replacement text corresponding to the symbol key:

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

symbolpluralaccess The replacement text corresponding to the symbolplural key:

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

descriptionaccess The replacement text corresponding to the description key:

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

descriptionpluralaccess The replacement text corresponding to the descriptionplural key:

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

shortaccess The replacement text corresponding to the short key:

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

shortpluralaccess The replacement text corresponding to the shortplural key:

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

longaccess The replacement text corresponding to the long key:

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

longpluralaccess The replacement text corresponding to the longplural key:

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

There are now also keys that correspond to the user keys:

user1access The replacement text corresponding to the user1 key:

```
\define@key{glossentry}{user1access}{%  
  \def\@glo@useriaccess{#1}%  
}
```

user2access The replacement text corresponding to the user2 key:

```
\define@key{glossentry}{user2access}{%  
  \def\@glo@useriiaccess{#1}%  
}
```

user3access The replacement text corresponding to the user3 key:

```
\define@key{glossentry}{user3access}{%  
  \def\@glo@useriiiaccess{#1}%  
}
```


`user4access` The replacement text corresponding to the `user4` key:

```
\define@key{glossentry}{user4access}{%
  \def\@glo@userivaccess{#1}%
}
```

`user5access` The replacement text corresponding to the `user5` key:

```
\define@key{glossentry}{user5access}{%
  \def\@glo@uservaccess{#1}%
}
```

`user6access` The replacement text corresponding to the `user6` key:

```
\define@key{glossentry}{user6access}{%
  \def\@glo@userviaccess{#1}%
}
```

For any custom keys, the replacement text would have to be explicitly put in the value, e.g., `user1={\glsshortaccsupp{inches}{in}}`.

Append these new keys to `\@gls@keymap`:

```
\appto\@gls@keymap{,%
  {access}{access},%
  {textaccess}{textaccess},%
  {firstaccess}{firstaccess},%
  {pluralaccess}{pluralaccess},%
  {firstpluralaccess}{firstpluralaccess},%
  {symbolaccess}{symbolaccess},%
  {symbolpluralaccess}{symbolpluralaccess},%

  {descriptionaccess}{descaccess},%
  {descriptionpluralaccess}{descpluralaccess},%
  {shortaccess}{shortaccess},%
  {shortpluralaccess}{shortpluralaccess},%
  {longaccess}{longaccess},%
  {longpluralaccess}{longpluralaccess},%
  {user1access}{useriaccess},%
  {user2access}{useriiaccess},%
  {user3access}{useriiiaccess},%
  {user4access}{userivaccess},%
  {user5access}{uservaccess},%
  {user6access}{userviaccess}%
}
```

`\@gls@noaccess` Indicates that no replacement text has been provided.

```
\def\@gls@noaccess{\relax}
```

Previously, the `access` key was initialised to the value of the `symbol` key at the start for backwards compatibility. This causes a problem for situations where the replacement text is provided for `symbol` but not for `name` so this behaviour has been removed.

```
\let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
```

```

\renewcommand*{\@newglossaryentryprehook}{%
  \@gls@oldnewglossaryentryprehook
\def\@glo@access{\relax}%

```

Initialise the other keys:

```

\def\@glo@textaccess{\@glo@access}%
\def\@glo@firstaccess{\@glo@access}%
\def\@glo@pluralaccess{\@glo@textaccess}%
\def\@glo@firstpluralaccess{\@glo@pluralaccess}%
\def\@glo@symbolaccess{\relax}%
\def\@glo@symbolpluralaccess{\@glo@symbolaccess}%
\def\@glo@descaccess{\relax}%
\def\@glo@descpluralaccess{\@glo@descaccess}%
\def\@glo@shortaccess{\relax}%
\def\@glo@shortpluralaccess{\@glo@shortaccess}%
\def\@glo@longaccess{\relax}%
\def\@glo@longpluralaccess{\@glo@longaccess}%
\def\@glo@useriaccess{\relax}%
\def\@glo@useriiaaccess{\relax}%
\def\@glo@useriiiaaccess{\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 @useriiaaccess\endcsname{%
    \@glo@useriiaaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiiaaccess\endcsname{%
    \@glo@useriiiaaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @userivaccess\endcsname{%
    \@glo@userivaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @uservaccess\endcsname{%
    \@glo@uservaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @userviaccess\endcsname{%
    \@glo@userviaccess}%
}

```

6.2 Accessing Replacement Text

`\glsentryaccess` Get the value of the access key for the entry with the given label:

```

\newcommand*{\glsentryaccess}[1]{%
    \@gls@entry@field{#1}{access}%
}

```

`\glsentrytextaccess` Get the value of the textaccess key for the entry with the given label:

```

\newcommand*{\glsentrytextaccess}[1]{%
    \@gls@entry@field{#1}{textaccess}%
}

```

}

`\glsentryfirstaccess` Get the value of the `firstaccess` key for the entry with the given label:

```
\newcommand*{\glsentryfirstaccess}[1]{%
  \@gls@entry@field{#1}{firstaccess}%
}
```

`\glsentrypluralaccess` Get the value of the `pluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentrypluralaccess}[1]{%
  \@gls@entry@field{#1}{pluralaccess}%
}
```

`\glsentryfirstpluralaccess` Get the value of the `firstpluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentryfirstpluralaccess}[1]{%
  \@gls@entry@field{#1}{firstpluralaccess}%
}
```

`\glsentrysymbolaccess` Get the value of the `symbolaccess` key for the entry with the given label:

```
\newcommand*{\glsentrysymbolaccess}[1]{%
  \@gls@entry@field{#1}{symbolaccess}%
}
```

`\glsentrysymbolpluralaccess` Get the value of the `symbolpluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentrysymbolpluralaccess}[1]{%
  \@gls@entry@field{#1}{symbolpluralaccess}%
}
```

`\glsentrydescaccess` Get the value of the `descriptionaccess` key for the entry with the given label:

```
\newcommand*{\glsentrydescaccess}[1]{%
  \@gls@entry@field{#1}{descaccess}%
}
```

`\glsentrydescpluralaccess` Get the value of the `descriptionpluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentrydescpluralaccess}[1]{%
  \@gls@entry@field{#1}{descpluralaccess}%
}
```

`\glsentryshortaccess` Get the value of the `shortaccess` key for the entry with the given label:

```
\newcommand*{\glsentryshortaccess}[1]{%
  \@gls@entry@field{#1}{shortaccess}%
}
```

`\glsentryshortpluralaccess` Get the value of the `shortpluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentryshortpluralaccess}[1]{%
  \@gls@entry@field{#1}{shortpluralaccess}%
}
```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

\newcommand*{\glentryuserviaccess}[1]{%
  \@gls@entry@field{#1}{userviaccess}%
}

```

There are three types of replacement text:

Alt Description of some content that's non-textual (for example, an image). A word break is assumed after the content.

ActualText A character or sequence of characters that replaces textual content (for example, a dropped capital, a ligature or a symbol). No word break is assumed after the content.

E Expansion of an abbreviation to avoid ambiguity (for example, “St” could be short for “saint” or “street”).

Therefore, rather than having one command for all fields, it’s better to have a command dependent on the field type. For example, the `short` and `shortpl` keys would require `E`, the `symbol` key would require `ActualText`, and a field that contains an image would require `Alt`.

```
\glsfieldaccsupp{<replacement>}{<content>}{<field>}{<label>}
```

`\glsfieldaccsupp`

Test if there’s a command called `\gls<field>accsupp`. If there is then use that otherwise use `\glsaccsupp`. The first argument should be the internal field label (not the key). The final argument is the entry label. If `glossaries-extra` has been loaded, this first checks for `\glsxtr<category><field>accsupp` and `\glsxtr<category>accsupp`.

```
\newcommand{\glsfieldaccsupp}[4]{%
  \ifdef\glscategory
  {%
    \ifcsdef{glsxtr\glscategory{#4}#3accsupp}%
    {\csname glsxtr\glscategory{#4}#3accsupp\endcsname{#1}{#2}}%
    {%
      \ifcsdef{glsxtr\glscategory{#4}accsupp}%
      {\csname glsxtr\glscategory{#4}accsupp\endcsname{#1}{#2}}%
      {%
        \ifcsdef{gls#3accsupp}%
        {\csname gls#3accsupp\endcsname{#1}{#2}}%
        {\glsaccsupp{#1}{#2}}%
      }%
    }%
  }%
  {%
    \ifcsdef{gls#3accsupp}%
    {\csname gls#3accsupp\endcsname{#1}{#2}}%
    {\glsaccsupp{#1}{#2}}%
  }%
}
```

```
\xglsfieldaccsupp{<replacement>}{<content>}{<field>}{<label>}
```

`\xglsfieldaccsupp`

As `\glsfieldaccsupp` but fully expand replacement text.

```
\newcommand{\xglsfieldaccsupp}[1]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsfieldaccsupp\expandafter{\@gls@replacementtext}%
}
```

`\glsshortaccsupp`

```
\glsshortaccsupp{<replacement text>}{<text>}
```

```
\newcommand*{\glsshortaccsupp}[2]{\glsaccessibility{E}{#1}{#2}}
```

`\glsshortplaccsupp`

```
\glsshortplaccsupp{<replacement text>}{<text>}
```

```
\newcommand*{\glsshortplaccsupp}{\glsshortaccsupp}
```

`\glsaccsupp`

```
\glsaccsupp{<replacement text>}{<text>}
```

```
\newcommand*{\glsaccsupp}[2]{\glsaccessibility{ActualText}{#1}{#2}}
```

`\xglsaccsupp` Fully expands replacement text before calling `\glsaccsupp`

```
\newcommand*{\xglsaccsupp}[2]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsaccsupp\expandafter{\@gls@replacementtext}{#2}%
}
```

`\@gls@access@display` Deprecated in v4.45. Use `\@gls@fieldaccess@display` instead.

```
\newcommand*{\@gls@access@display}[2]{%
  \protected@edef\@glo@access{#2}%
  \ifx\@glo@access\@gls@noaccess
    #1%
  \else
    \xglsaccsupp{\@glo@access}{#1}%
  \fi
}
```

`\@gls@fieldaccess@display`

```
\@gls@fieldaccess@display{<label>}{<field>}{<content>}{<replacement>}
```

```
\newcommand*{\@gls@fieldaccess@display}[4]{%
  \protected@edef\@glo@access{#4}%
  \ifdefequal\@glo@access\@gls@noaccess
    {#3}%
  {\expandafter\glsfieldaccsupp\expandafter{\@glo@access}{#3}{#2}{#1}}%
}
```

`\glsnameaccessdisplay` Displays the first argument with the accessibility text for the entry with the label given by the second argument (if set).

```
\newrobustcmd*{\glsnameaccessdisplay}[2]{%
  \ifcsundef{glo@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}}}%
    }

\glsdescriptionpluralaccessdisplay 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}{\glentryuseriiaccess{#2}}}%
}

```

`\glsuseriiiaccessdisplay` As above but for the user3access replacement text.

```

\newrobustcmd*{\glsuseriiiaccessdisplay}[2]{%
  \ifcsundef{glo@glstoklabel{#2}@useriiiaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useriii}{#1}{\glentryuseriiiaccess{#2}}}%
}

```

`\glsuserivaccessdisplay` As above but for the user4access replacement text.

```

\newrobustcmd*{\glsuserivaccessdisplay}[2]{%
  \ifcsundef{glo@glstoklabel{#2}@userivaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useriv}{#1}{\glentryuserivaccess{#2}}}%
}

```

`\glsuservaccessdisplay` As above but for the user5access replacement text.

```

\newrobustcmd*{\glsuservaccessdisplay}[2]{%
  \ifcsundef{glo@glstoklabel{#2}@uservaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{userv}{#1}{\glentryuservaccess{#2}}}%
}

```

`\glsuserviaccessdisplay` As above but for the user6access replacement text.

```

\newrobustcmd*{\glsuserviaccessdisplay}[2]{%
  \ifcsundef{glo@glstoklabel{#2}@userviaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{uservi}{#1}{\glentryuserviaccess{#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

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

Singular form

```
\glscapscale
{%
```

Don't adjust case

```
\ifglused\glslabel
{%
```

Subsequent use

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

First use

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

Make first letter upper case

```
\ifglused\glslabel
{%
```

Subsequent use

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

First use

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

```

    }%
    {%
Make all upper case
    \ifglused\glslabel
    {%
Subsequent use
    \glstextaccessdisplay
    {\glsupercase{\glentrytext{\glslabel}}}{\glslabel}%
    \glsupercase{\glsinsert}%
    }%
    {%
First use
    \glsfirstaccessdisplay
    {\glsupercase{\glentryfirst{\glslabel}}}{\glslabel}%
    \glsupercase{\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
    {%
        \ifglused\glslabel
        {%
Subsequent use:
            \glsifplural
            {%
Subsequent plural form:
                \glscapscase
                {%
Subsequent plural form, don't adjust case:
                    \acronymfont
                    {\glsshortpluralaccessdisplay
                     {\glentryshortpl{\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:

```
\genplacrformat{\glslabel}{\glsinsert}%
}%
{%
```

First use plural form, make first letter upper case:

```
\Genplacrformat{\glslabel}{\glsinsert}%
}%
{%
```

First use plural form, all caps:

```
\glsuppercase
{\genplacrformat{\glslabel}{\glsinsert}}%
}%
}%
{%
```

First use singular form

```
\glscapscase
{%
```

First use singular form, don't adjust case:

```
\genacrformat{\glslabel}{\glsinsert}%
}%
{%
```

First use singular form, make first letter upper case:

```
\Genacrformat{\glslabel}{\glsinsert}%
}%
{%
```

First use singular form, all caps:

```
\glsuppercase
{\genacrformat{\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
}%
}
```

`\genacrformat` Redefine to include accessibility information.

```
\renewcommand*\genacrformat}[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@gl@link@checkfirsthyper\relax
        \let\glsifplural\@secondoftwo
        \let\glscapscase\@firstofthree
        \let\glsinsert\@empty
        \def\glscustomtext{%
            \acronymfont{\glsshortaccessdisplay{\glsentryshort{#2}}{#2}}#3%
        }%
    }

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

\glspostlinkhook
}

\@Acrshort
\def\@Acrshort#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gl@link@checkfirsthyper\relax
        \let\glsifplural\@secondoftwo
        \let\glscapscase\@secondofthree
        \let\glsinsert\@empty
        \def\glscustomtext{%
            \acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
        }%
    }

```



```

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{\glstryshort{#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{\glstrylong{#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{\glsupercase{\glsentryfirst{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsplural@
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\glsentryplural{#2}}{#2}#3}%
}

\@Glsplural@
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\Glsentryplural{#2}}{#2}#3}%
}

\@GLSplural@
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glspluralaccessdisplay{\glsupercase{\glsentryplural{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsfirstplural@
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\glsentryfirstplural{#2}}{#2}#3}%
}

\@Glsfirstplural@
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\Glsentryfirstplural{#2}}{#2}#3}%
}

\@GLSfirstplural@
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsfirstpluralaccessdisplay{\glsupercase{\glsentryfirstplural{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

```

```

\@glsname@
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glsentryname{#2}}{#2}#3}%
}

\@Glsname@
\def\@Glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\Glsentryname{#2}}{#2}#3}%
}

\@GLSname@
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsnameaccessdisplay{\glsuppercase{\glsentryname{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

\@glsdesc@
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\glsentrydesc{#2}}{#2}#3}%
}

\@Glsdesc@
\def\@Glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
}

\@GLSdesc@
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsdescriptionaccessdisplay{\glsuppercase{\glsentrydesc{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

\@glsdescplural@
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\glsentrydescplural{#2}}{#2}#3}%
}

\@Glsdescplural@
\def\@Glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
}

\@GLSdescplural@
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsdescriptionpluralaccessdisplay{\glsuppercase{\glsentrydescplural{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

```

```

\@glssymbol@
\def\@glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\glentrysymbol{#2}}{#2}#3}%
}

\@Glsymbol@
\def\@Glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}

\@GLSymbol@
\def\@GLSymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glssymbolaccessdisplay{\glsupercase{\glentrysymbol{#2}}}{#2}%
    \glsupercase{#3}}}%
}

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

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

\@GLSymbolplural@
\def\@GLSymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glssymbolpluralaccessdisplay{\glsupercase{\glentrysymbolplural{#2}}}{#2}%
    \glsupercase{#3}}}%
}

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

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

\@GLSuseri@
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriaccessdisplay{\glsupercase{\glentryuseri{#2}}}{#2}%
    \glsupercase{#3}}}%
}

```

```

\@glsuserii@
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glsentryuserii{#2}}{#2}#3}%
}

\@Glsuserii@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glsentryuserii{#2}}{#2}#3}%
}

\@GLSuserii@
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiaccessdisplay{\glsupercase{\glsentryuserii{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsuseriii@
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\glsentryuseriii{#2}}{#2}#3}%
}

\@Glsuseriii@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glsentryuseriii{#2}}{#2}#3}%
}

\@GLSuseriii@
\def\@GLSuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiiaccessdisplay{\glsupercase{\glsentryuseriii{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsuseriv@
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glsentryuseriv{#2}}{#2}#3}%
}

\@Glsuseriv@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
}

\@GLSuseriv@
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuserivaccessdisplay{\glsupercase{\glsentryuseriv{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

```

```

\@glsuserv@
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\glsentryuserv{#2}}{#2}#3}%
}

\@Glsuserv@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\Glsentryuserv{#2}}{#2}#3}%
}

\@GLSuserv@
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuservaccessdisplay{\glsupercase{\glsentryuserv{#2}}}{#2}%
    \glsupercase{#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{\glsupercase{\glsentryuservi{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

```

6.3 Displaying the Glossary

We need to redefine the way the glossary entries are formatted to include the accessibility support. The predefined glossary styles use `\glossentryname`, `\glossentrydesc` and `\glossentrysymbol`, but we need to provide compatibility with earlier versions in case users have defined their own styles using `\accsuppglossaryentryfield` and `\accsuppglossarysubentryfield`.

Now redefine `\glossentryname`, `\glossentrydesc` and `\glossentrysymbol` etc so they use the accessibility stuff.

```

\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\glsentryname{#1}}{#1}}%
  }%
}

```

```

\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\Glsentryname{#1}}{#1}}%
  }%
}

\renewcommand*{\glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}

\renewcommand*{\Glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}

\renewcommand*{\glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
  }%
}

\renewcommand*{\Glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
  }%
}

```

\accsuppglossaryentryfield

```

\newcommand*{\accsuppglossaryentryfield}[5]{%
  \glossaryentryfield{#1}%
  {\glsnameaccessdisplay{#2}{#1}}%
  {\glsdescriptionaccessdisplay{#3}{#1}}%
  {\glssymbolaccessdisplay{#4}{#1}}{#5}%
}

```

\accsuppglossarysubentryfield

```

\newcommand*{\accsuppglossarysubentryfield}[6]{%
  \glossarysubentryfield{#1}{#2}%
  {\glsnameaccessdisplay{#3}{#2}}%
  {\glsdescriptionaccessdisplay{#4}{#2}}%
  {\glssymbolaccessdisplay{#5}{#2}}{#6}%
}

```


6.4 Acronyms

Redefine acronym styles provided by glossaries:

long-short $\langle long \rangle$ ($\langle short \rangle$) acronym style.

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

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

```
\ifglshaslong{\glslabel}{\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{\Glseyentrylong{##1}}{##1}##2\space
\glsshortaccessdisplay
{\protect\firstacronymfont{\glseyentryshort{##1}}{##1})}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glslongpluralaccessdisplay{\glseyentrylongpl{##1}}{##1}##2\space
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glseyentryshortpl{##1}}{##1})}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\glslongpluralaccessdisplay{\Glseyentrylongpl{##1}}{##1}##2\space
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glseyentryshortpl{##1}}{##1})}%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glseyentryshort{##1}}{##1}}
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
```

short-long $\langle short \rangle$ ($\langle long \rangle$) acronym style.

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

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

```
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glslongaccessdisplay{\glseyentrylong{##1}}{##1}##2\space
\glsshortaccessdisplay
{\protect\firstacronymfont{\glseyentryshort{##1}}{##1})}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glslongpluralaccessdisplay{\glseyentrylongpl{##1}}{##1}##2\space
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glseyentryshortpl{##1}}{##1})}%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glseyentryshort{##1}}{##1}}
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
```

```

\glsshortaccessdisplay
  {\protect\firstacronymfont{\glentryshort{##1}}}{##1}##2\space
(\glslongaccessdisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
  \glsshortaccessdisplay
    {\protect\firstacronymfont{\Glentryshort{##1}}}{##1}##2\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
  \glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\glentryshortpl{##1}}}{##1}##2\space
  (\glslongpluralaccessdisplay
    {\glentrylongpl{##1}}{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\Glentryshortpl{##1}}}{##1}##2\space
  (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}

```

`long-short-desc` $\langle long \rangle$ ($\{\langle short \rangle\}$) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{long-short-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1})}%
  }

```

`long-sc-short-desc` $\langle long \rangle$ ($\backslash\text{textsc}\{\langle short \rangle\}$) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{long-sc-short-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{long-sc-short}%
}%

```

```

{
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}

long-sm-short-desc <long> (\textsmaller{<short>}) acronym style that has an accompanying de-
scription (which the user needs to supply).
\renewacronymstyle{long-sm-short-desc}%
{
  \GlsUseAcrEntryDisplayStyle{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}%
{
  \GlsUseAcrEntryDisplayStyle{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}%
{
  \GlsUseAcrEntryDisplayStyle{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
{\glsupercase{\glsentrylongpl{\glslabel}}}{\glslabel}%

```

```

        \glssupercase{\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
        {\glssupercase
         {\glsentrylong{\glslabel}\glsinsert}}{\glslabel}%
        \glssupercase{\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
        {\glssupercase{\glsentrylong{##2}}{##2}##3\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
    \glslink[##1]{##2}{%

```

```

\glslongpluralaccessdisplay
  {\glsentrylongpl{##2}}{##2}##3\space
(\glsshortpluralaccessdisplay
  {\acronymfont{\glsentryshortpl{##2}}{##2}})%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\Glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
      {\acronymfont{\glsentryshortpl{##2}}{##2}})%
  }%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\glsupercase{\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
      {\acronymfont{\glsentryshortpl{##2}}{##2}})%
  }%
\renewcommand*{\glsentryfull}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
  \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}\space
  (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space
  (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}{##1}})%
}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}

```

dua-desc *<long>* only acronym style with user-supplied description.

```

\renewacronymstyle{dua-desc}%
{%
  \GlsUseAcrEntryDispStyle{dua}%
}%
{%
  \GlsUseAcrStyleDefs{dua}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\acronymfont{\glsentrylong{##1}}{##1}}%
\renewcommand*{\acronymsort}[2]{##2}%

```

}%

footnote *<short>*\footnote{*<long>*} acronym style.

\renewacronymstyle{footnote}%

{%

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

\ifglshaslong{\glslabel}{\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
  {\glssupercase
    {\acronymfont{\glentryshort{##2}}}{##2}##3\space
    (\glslongaccessdisplay{\glentrylong{##2}}{##2}))}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\glentryshortpl{##2}}}{##2}##3\space
      (\glslongpluralaccessdisplay{\glentrylongpl{##2}}{##2}))}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\Glsentryshortpl{##2}}}{##2}##3\space
      (\glslongpluralaccessdisplay{\glentrylongpl{##2}}{##2}))}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\glssupercase
        {\acronymfont{\glentryshortpl{##2}}}{##2}##3\space
        (\glslongpluralaccessdisplay{\glentrylongpl{##2}}{##2}))}%

```

Similarly for \glentryfull etc:

```

\renewcommand*{\glentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1}))%
\renewcommand*{\Glsentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##1}}}{##1}\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1}))%
\renewcommand*{\glentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\glentryshortpl{##1}}}{##1}\space
    (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1}))%
\renewcommand*{\Glsentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\Glsentryshortpl{##1}}}{##1}\space
    (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1}))%
}

```

footnote-sc \textsc{<short>}\footnote{<long>} acronym style.

```

\renewacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDisplayStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##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}})%
}
```

```
\glsglsdefaultshortaccess{\<long>}{\<short>}
```

`\glsglsdefaultshortaccess`

Default shortaccess value.

```
\newcommand*{\glsglsdefaultshortaccess}[2]{#1}
```

Use `\newacronymhook` to modify the key list to set the access text to the long version by default.

```
\renewcommand*{\newacronymhook}{%
\protected@edef\@glsgls@keylist{%
shortaccess={\glsglsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
shortpluralaccess={\glsglsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
\the\glsgls@keylisttok}%
\expandafter\glsgls@keylisttok\expandafter{\@glsgls@keylist}%
}
```

`\DefaultNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\DescriptionFootnoteNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\DescriptionNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\FootnoteNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\SmallNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

The following are kept for compatibility with versions before 3.0:

`\glsshortaccesskey`

```
\newcommand*{\glsshortaccesskey}{\glsshortkey access}%
```

`\glsshortpluralaccesskey`

```
\newcommand*{\glsshortpluralaccesskey}{\glsshortpluralkey access}%
```

`\glslongaccesskey`

```
\newcommand*{\glslongaccesskey}{\glslongkey access}%
```

```
\glslongpluralaccesskey
\newcommand*{\glslongpluralaccesskey}{\glslongpluralkey access}%
```

6.5 Debugging Commands

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

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

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

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

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

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

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

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

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

```

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

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

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

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

```

7 Backwards Compatibility

7.1 glossaries-compatible-207

Provides compatibility with version 2.07 and below. This uses original `glossaries` 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{v4.51}{2022-11-02}

```

Identify package.

```
\ProvidesPackage{glossaries-compatible-207}[2022/11/02 v4.51 (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{v4.51}{2022-11-02}
```

Identify package.

```
\ProvidesPackage{glossaries-compatible-307}[2022/11/02 v4.51 (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]{%
\glsshowtargetsymbols\marginpar{\glsshowtargetsymbols\glsshowtargetfont #1}}
\newcommand*{\glsshowtargetsymbols}{\tiny$\triangleright$}
\newcommand*{\glsshowtargetfont}{\ttfamily\footnotesize}
\newcommand*{\@glsshowtarget}[1]{%
\newrobustcmd*{\@glsshowtarget}[1]{%
\begingroup
\protected@edef\@gls@tmp{#1}%
\@onelevel@sanitize\@gls@tmp
\expandafter\glsshowtarget\expandafter{\@gls@tmp}%
\endgroup
}
\newcommand*{\@glsshowaccsupp}[3]{%
\newrobustcmd*{\glsshowaccsupp}[3]{%
\ifstrempy{#1}%
{\@glsshowtarget{/#2 (#3)}}%
{\@glsshowtarget{/#2 (#3) [#1]}}%
}
}
\newcommand*{\@gls@see@noindex}{%
\PackageError{glossaries}%
{'\gls@xr@key' key may only be used after \string\makeglossaries\space
or \string\makenoidxglossaries\space (or move
\string\newglossaryentry\space
definitions into the preamble)}}%
{You must use \string\makeglossaries\space
or \string\makenoidxglossaries\space before defining
any entries that have a '\gls@xr@key' key. It may
be that the 'see' key has been written to the .glsdefs
file from the previous run, in which case you need to
move your definitions
to the preamble if you don't want to use
\string\makeglossaries\space
or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoinindex}%
[\gls@seenoinindex@val\gls@seenoinindex@nr]{error,warn,ignore}{%
\ifcase\gls@seenoinindex@nr
\renewcommand*{\@gls@see@noindex}{%
\PackageError{glossaries}%
{'\gls@xr@key' key may only be used after \string\makeglossaries\space
or \string\makenoidxglossaries}%
{You must use \string\makeglossaries\space
or \string\makenoidxglossaries\space before defining
any entries that have a '\gls@xr@key' key}%
}%
}

```

```

\or
\renewcommand*{\@gls@see@noindex}{%
\GlossariesWarning{'\gls@xr@key' key ignored}%
}%
\or
\renewcommand*{\@gls@see@noindex}{}%
\fi
}
\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}%
{
\newcommand*{\@@glossarysec}{section}}%
{
\newcommand*{\@@glossarysec}{chapter}}
\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
subsection,subsubsection,paragraph,subparagraph}[section]{%
\renewcommand*{\@@glossarysec}{#1}}
\newcommand*{\@@glossarysecstar}{*}%
\newcommand*{\@@glossaryseclabel}{}%
\newcommand*{\glsautoprefix}{}%
\define@choicekey{glossaries.sty}{numberedsection}%
[\gls@numberedsection@val\gls@numberedsection@nr]{%
false,nolabel,autolabel,nameref}[nolabel]{%
\ifcase\gls@numberedsection@nr\relax
\renewcommand*{\@@glossarysecstar}{*}%
\renewcommand*{\@@glossaryseclabel}{}%
\or
\renewcommand*{\@@glossarysecstar}{}%
\renewcommand*{\@@glossaryseclabel}{}%
\or
\renewcommand*{\@@glossarysecstar}{}%
\renewcommand*{\@@glossaryseclabel}{}%
\label{\glsautoprefix\@glo@type}}%
\or
\renewcommand*{\@@glossarysecstar}{*}%
\renewcommand*{\@@glossaryseclabel}{}%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}}%
\fi
}
\@ifpackageloaded{classicthesis}
{
\newcommand*{\@glossary@default@style}{index}}
{
\newcommand*{\@glossary@default@style}{list}}
\define@key{glossaries.sty}{style}{%
\def\@glossary@default@style{#1}%
}
\newcommand*{\@gls@declareoption}[2]{%
\DeclareOptionX{#1}{#2}%
\DeclareOption{#1}{#2}%
}
\newcommand*{\glossaryentrynumbers}[1]{#1\gls@save@numberlist{#1}}

```



```

\@gls@declareoption{nonumberlist}{%
  \renewcommand*{\glossaryentrynumbers}[1]{\gls@save@numberlist{#1}}%
}
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{}
\gls@savenumberlistfalse
\newcommand*\@glo@seeautonumberlist{}
\@gls@declareoption{seeautonumberlist}{%
  \renewcommand*{\@glo@seeautonumberlist}{%
    \def\@glo@prefix{\glsnextpages}%
  }%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{}
\gls@esclocationstrue
\newcommand*\@gls@loadlong{\RequirePackage{glossary-long}[=v4.46]}
\@gls@declareoption{nolong}{\renewcommand*{\@gls@loadlong}{}%
\IfFileExists{supertabular.sty}{%
  \newcommand*\@gls@loadsuper{\RequirePackage{glossary-super}[=v4.46]}{}%
  \newcommand*\@gls@loadsuper{}}%
\@gls@declareoption{nosuper}{\renewcommand*{\@gls@loadsuper}{}%
\newcommand*\@gls@loadlist{\RequirePackage{glossary-list}[=v4.46]}%
\@gls@declareoption{nolist}{%
  \renewcommand*{\@gls@loadlist}{%
    \ifdefstring{\@glossary@default@style}{list}%
    {\let\@glossary@default@style\relax}%
  }%
}%
}
\newcommand*\@gls@loadtree{\RequirePackage{glossary-tree}[=v4.46]}
\@gls@declareoption{notree}{\renewcommand*{\@gls@loadtree}{}%
\@gls@declareoption{nostyles}{%
  \renewcommand*{\@gls@loadlong}{}%
  \renewcommand*{\@gls@loadsuper}{}%
  \renewcommand*{\@gls@loadlist}{}%
  \renewcommand*{\@gls@loadtree}{}%
  \let\@glossary@default@style\relax
}
\newcommand*\@gls@postdescription{%
  \ifgls@postdot\else.\spacefactor\sfcode'\. \fi
}
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{}
\gls@nopostdotfalse
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{}
\gls@nogroupskipfalse
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{}
\@ifclassloaded{memoir}{%
  \gls@ucmarktrue
}%
{%
  \gls@ucmarkfalse
}

```

```

}
\newcommand*{\@gls@define@glossaryentrycounter}{%
  \ifglentrycounter
    \ifundef\c@glossaryentry
    {%
      \ifx\@gls@counterwithin\@empty
        \newcounter{glossaryentry}%
      \else
        \newcounter{glossaryentry}[\@gls@counterwithin]%
      \fi
      \def\theHglossaryentry{\currentglossary.\theglossaryentry}%
    }%
  }%
\fi
}
\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\glentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
  \renewcommand*{\@gls@counterwithin}{#1}%
  \glentrycountertrue
  \@gls@define@glossaryentrycounter
}
\newcommand*{\@gls@counterwithin}{}
\newcommand*{\@gls@define@glossarysubentrycounter}{%
  \ifundef\c@glossarysubentry
  {%
    \ifglssubentrycounter
      \ifglentrycounter
        \newcounter{glossarysubentry}[glossaryentry]%
      \else
        \newcounter{glossarysubentry}%
      \fi
      \def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
    \fi
  }%
  {}%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
\glssubentrycounterfalse
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none}{%
  \renewcommand*{\@glo@default@sorttype}{#1}%
  \csname @gls@setupsort@#1\endcsname
}
\newcommand*{\glsprestandardsort}[3]{%
  \glsdosanitizesort
}
\newcommand*{\@glo@check@sortallowed}[1]{}
\newcommand*{\@gls@setupsort@standard}{%
  \def\do@glo@storeentry{\@glo@storeentry}%

```

```

\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
  \ifx\@glo@sort\@glsdefaultsort
    \let\@glo@sort\@glo@name
  \fi
  \let\glsdosanitizesort\@gls@sanitizesort
  \glsprestandardsort{\@glo@sort}{##1}{##2}%
  \expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\@gls@setupsort@standard
\newcommand*\glsortnumberfmt[1]{%
  \ifnum#1<100000 0\fi
  \ifnum#1<10000 0\fi
  \ifnum#1<1000 0\fi
  \ifnum#1<100 0\fi
  \ifnum#1<10 0\fi
  \number#1%
}
\newcommand*\@gls@setupsort@def{%
\def\do@glo@storeentry{\@glo@storeentry}%
\def\@gls@defsortcount##1{%
  \expandafter\global
  \expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
  \ifcsundef{glossary@##1@sortcount}%
  {\@gls@defsortcount{##1}}%
  {}%
  \expandafter\global\expandafter
  \advance\csname glossary@##1@sortcount\endcsname by 1\relax
  \expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
    \expandafter\glsortnumberfmt
    {\csname glossary@##1@sortcount\endcsname}}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*\@gls@setupsort@use{%
\let\do@glo@storeentry\@gobble
\def\@gls@defsortcount##1{%
  \expandafter\global
  \expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
  \expandafter\gdef\csname glo@##2@sort\endcsname{}%
}%
\def\@gls@setsort##1{%

```

```

\edef\@glo@parent{\csname glo@##1@parent\endcsname}%
\ifx\@glo@parent\@empty
\else
\expandafter\@gls@setsort\expandafter{\@glo@parent}%
\fi
\edef\@glo@type{\csname glo@##1@type\endcsname}%
\edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
\ifx\@gls@tmp\@empty
\expandafter\global\expandafter
\advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
\expandafter\protected\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@gls@isacronymlist
\newcommand*{\gls@checkisacronymlist}[1]{%
  \glsIfListOfAcronyms{#1}%
  {\@gls@isacronymlisttrue}{\@gls@isacronymlistfalse}%
}
\newcommand*{\SetAcronymLists}[1]{%
  \renewcommand*{\@gls@acronymlists}{#1}%
}
\define@key{glossaries.sty}{acronymlists}{%
  \DeclareAcronymList{#1}%
}
\newcommand{\glscounter}{page}
\define@key{glossaries.sty}{counter}{%
  \renewcommand*{\glscounter}{#1}%
}
\newcommand*{\@gls@nohyperlist}{}
\newcommand*{\GlsDeclareNoHyperList}[1]{%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*{\@gls@nohyperlist}{#1}%
  }%
  {%
    \appto\@gls@nohyperlist{,#1}%
  }%
}
\define@key{glossaries.sty}{nohypertypes}{%
  \GlsDeclareNoHyperList{#1}%
}
\newcommand*{\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{#1}%
}
\newcommand*{\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{#1}%
}
\newcommand{\glosortentrieswarning}{%
  \typeout{Using TeX to sort glossary entries---this may
  take a while}%
}
\@gls@declareoption{nowarn}{%
  \if@gls@debug
    \GlossariesWarning{Warnings can't be suppressed in debug mode}%
  \else
    \renewcommand*{\GlossariesWarning}[1]{}%
  \fi
}

```

```

\renewcommand*\GlossariesWarningNoLine}[1]{}%
\renewcommand*\glosortentrieswarning{}%
\renewcommand*\@gls@missinglang@warn}[2]{}%
\fi
}
\newcommand*\@gls@missinglang@warn}[2]{%
\PackageWarningNoLine{glossaries}%
{No language module detected for ‘#1’.\MessageBreak
Language modules need to be installed separately.\MessageBreak
Please check on CTAN for a bundle called\MessageBreak
‘glossaries-#2’ or similar}%
}
\@gls@declareoption{nolangwarn}{%
\renewcommand*\@gls@missinglang@warn}[2]{}%
}
\newcommand*\@gls@warnonglossdefined{%
\GlossariesWarning{Overriding \string\printglossary}%
}
\newcommand*\@gls@warnontheGLOSSdefined{%
\GlossariesWarning{Overriding ‘theGLOSS’ environment}%
}
\@gls@declareoption{noredefwarn}{%
\renewcommand*\@gls@warnonglossdefined{}%
\renewcommand*\@gls@warnontheGLOSSdefined{}%
}
\newcommand*\@gls@sanitizedesc{%
}
\newcommand*\glssetexpandfield}[1]{%
\csdef{gls@assign@#1@field}##1##2{%
\@gls@expand@field{##1}{#1}{##2}%
}%
}
\newcommand*\glssetnoexpandfield}[1]{%
\csdef{gls@assign@#1@field}##1##2{%
\@gls@noexpand@field{##1}{#1}{##2}%
}%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}
\glssetnoexpandfield{descplural}
\newcommand*\@gls@sanitizename{}
\glssetnoexpandfield{name}
\newcommand*\@gls@sanitizesymbol{}
\glssetnoexpandfield{symbol}
\glssetnoexpandfield{symbolplural}
\newcommand*\@gls@sanitizesort{%
\ifgls@sanitizesort
\@gls@sanitizesort
\else
\@gls@nosanitizesort

```

```

\fi
}
\newcommand*{\@@gls@sanitizesort{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@@gls@nosanitizesort}{%
\newcommand*{\@gls@noidx@sanitizesort{%
  \ifdefvoid\@glo@sort
  {}%
  {%
    \expandafter\@@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
  \def\@glo@sort{#1#2}%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@@gls@noidx@nosanitizesort}{%
  \ifdefvoid\@glo@sort
  {}%
  {%
    \expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
  \bgroup
  \glsnoidxstripaccents
  \protected@xdef\@@glo@sort{#1#2}%
  \egroup
  \let\@glo@sort\@@glo@sort
}
\newcommand*{\glsnoidxstripaccents{%
  \let\IeC\@firstofone
  \let\add@accent@\@secondoftwo
  \let\@text@composite@x\@secondoftwo
  \let\@tabacckludge\@secondoftwo
  \expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
  \let\'\@firstofone
  \let\'\@firstofone
  \let\~\@firstofone
  \let\~\@firstofone
  \let\u\@firstofone
  \let\t\@firstofone
  \let\d\@firstofone
  \let\r\@firstofone
  \let=\@firstofone
  \let\.\@firstofone

```



```

\let~\@firstofone
\let\v\@firstofone
\let\H\@firstofone
\let\c\@firstofone
\let\b\@firstofone
\let\a\@secondoftwo
\def\AE{AE}%
\def\ae{ae}%
\def\OE{OE}%
\def\oe{oe}%
\def\AA{AA}%
\def\aa{aa}%
\def\L{L}%
\def\l{l}%
\def\O{O}%
\def\o{o}%
\def\SS{SS}%
\def\ss{ss}%
\def\th{th}%
\def\TH{TH}%
\def\dh{dh}%
\def\DH{DH}%
}
\@ifl@t@r\fmtversion{2019/10/01}
{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}
\define@boolkey[gls]{sanitize}{description}[true]{%
\GlossariesWarning{sanitize={description} package option deprecated}%
\ifgls@sanitize@description
\glssetnoexpandfield{desc}%
\glssetnoexpandfield{descplural}%
\else
\glssetexpandfield{desc}%
\glssetexpandfield{descplural}%
\fi
}
\define@boolkey[gls]{sanitize}{name}[true]{%
\GlossariesWarning{sanitize={name} package option deprecated}%
\ifgls@sanitize@name
\glssetnoexpandfield{name}%
\else
\glssetexpandfield{name}%
\fi
}
\define@boolkey[gls]{sanitize}{symbol}[true]{%
\GlossariesWarning{sanitize={symbol} package option deprecated}%
\ifgls@sanitize@symbol
\glssetnoexpandfield{symbol}%

```

```

        \glsssetnoexpandfield{symbolplural}%
    \else
        \glsssetexpandfield{symbol}%
        \glsssetexpandfield{symbolplural}%
    \fi
}
\define@boolkey{glossaries.sty}[gls]{sanitizesort}[true]{%
    \ifglsssanitizesort
        \glsssetnoexpandfield{sortvalue}%
        \renewcommand*{\@gls@noidx@setsanitizesort}{%
            \glsssanitizesorttrue
            \glsssetnoexpandfield{sortvalue}%
        }%
    \else
        \glsssetexpandfield{sortvalue}%
        \renewcommand*{\@gls@noidx@setsanitizesort}{%
            \glsssanitizesortfalse
            \glsssetexpandfield{sortvalue}%
        }%
    \fi
}
\glsssanitizesorttrue
\glsssetnoexpandfield{sortvalue}%
\newcommand*{\@gls@noidx@setsanitizesort}{%
    \glsssanitizesortfalse
    \glsssetexpandfield{sortvalue}%
}
\define@choicekey[gls]{sanitize}{sort}{true,false}[true]{%
    \setbool{glsssanitizesort}{#1}%
    \ifglsssanitizesort
        \glsssetnoexpandfield{sortvalue}%
    \else
        \glsssetexpandfield{sortvalue}%
    \fi
    \GlossariesWarning{sanitize={sort} package option
        deprecated. Use sanitizesort instead}%
}
\define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
    \ifthenelse{\equal{#1}{none}}{%
        {%
            \GlossariesWarning{sanitize package option deprecated}%
            \glsssetexpandfield{name}%
            \glsssetexpandfield{symbol}%
            \glsssetexpandfield{symbolplural}%
            \glsssetexpandfield{desc}%
            \glsssetexpandfield{descplural}%
        }%
    }%
    \setkeys[gls]{sanitize}{#1}%
}%

```

```

}
\newif\ifglstranslate
\newcommand*\@gls@usetranslator{%
  \ifpackageloaded{polyglossia}%
  {%
    \let\glsifusetranslator\@secondoftwo
  }%
  {%
    \@ifpackageloaded{babel}%
    {%
      \IfFileExists{translator.sty}%
      {%
        \RequirePackage{translator}%
        \let\glsifusetranslator\@firstoftwo
      }%
    }%
  }%
  {}%
}%
}
\newcommand{\glsifusedtranslatordict}[3]{%
  \glsifusetranslator
  {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
  {#3}%
}
\@gls@declareoption{nottranslate}{%
  \glstranslatefalse
  \let\@gls@usetranslator\relax
  \let\glsifusetranslator\@secondoftwo
}
\define@choicekey{glossaries.sty}{translate}%
[\gls@translate@val\gls@translate@nr]%
{true,false,babel}[true]%
{%
  \ifcase\gls@translate@nr\relax
  \glstranslatetrue
  \renewcommand*\@gls@usetranslator{%
    \@ifpackageloaded{polyglossia}%
    {%
      \let\glsifusetranslator\@secondoftwo
    }%
    {%
      \@ifpackageloaded{babel}%
      {%
        \IfFileExists{translator.sty}%
        {%
          \RequirePackage{translator}%
          \let\glsifusetranslator\@firstoftwo
        }%
      }%
    }%
  }%
}

```

```

        }%
        {}%
    }%
}%
\or
\glstranslatefalse
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\or
\glstranslatetrue
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\fi
}
\glstranslatefalse
\let\glsifusetranslator\@secondoftwo
\@ifpackageloaded{translator}%
{%
\glstranslatetrue
\let\glsifusetranslator\@firstoftwo
}%
{%
\@for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
{
\@ifpackageloaded{\gls@thissty}%
{%
\glstranslatetrue
\endfortrue
}%
}%
}
}
\define@boolkey{glossaries.sty}[gls]{indexonlyfirst}[true]{}
\glsindexonlyfirstfalse
\define@boolkey{glossaries.sty}[gls]{hyperfirst}[true]{}
\glshyperfirsttrue
\newcommand*{\@gls@setacrstyle}{}
\define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{}
\ifbool{glsacrdescription}%
{%
}%
\renewcommand*{\@gls@sanitizedesc}{}%
}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{description}[true]{}
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{}

```

```

\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{dua}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{shortcuts}[true]{}
\newcommand*{\glsorder}{word}
\newcommand*{\@glsorder}[1]{}
\define@choicekey{glossaries.sty}{order}{word,letter}{%
\def\glsorder{#1}}
\newif\ifglxindy
\glxindyfalse
\@gls@declareoption{makeindex}{\glxindyfalse}
\define@boolkey[gls]{xindy}{glsnumbers}[true]{}
\gls@xindy@glsnumberstrue
\def\@xdy@main@language{\language}%
\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}[]{%
\glxindytrue
\setkeys[gls]{xindy}{#1}%
}
\@gls@declareoption{xindygloss}{%
\glxindytrue
}
\@gls@declareoption{xindynoglsnumbers}{%
\glxindytrue
\gls@xindy@glsnumbersfalse
}
\providecommand{\@domakeglossaries}[1]{#1}
\@gls@declareoption{disablemakegloss}{%
\ifdefequal\makeglossaries\@no@makeglossaries
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect
(\string\makenoidxglossaries\space has already been used)}}%
}%
{%
\ifdefequal\@makeglossary\@gobble
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect

```

```

        (\string\makeglossaries\space has already been used))}%
    }%
    {%
        \renewcommand{\@domakeglossaries}[1]{%
            \PackageInfo{glossaries}{\string\makeglossaries\space and
            \string\makenoidxglossaries\space have been disabled}%
        }%
    }%
}%
}
\@gls@declareoption{restoremakegloss}{%
    \ifdefequal\makeglossaries\@no@makeglossaries
    {%
        \GlossariesWarning{Option 'restoremakegloss' has no effect
        (\string\makenoidxglossaries\space has already been used))}%
    }%
    {%
        \ifdefequal\@makeglossary\@gobble
        {%
            \GlossariesWarning{Option 'restoremakegloss' has no effect
            (\string\makeglossaries\space has already been used))}%
        }%
        {%
            \PackageInfo{glossaries}{\string\makeglossaries\space and
            \string\makenoidxglossaries\space have been restored}%
            \let\@domakeglossaries\@firstofone
        }%
    }%
}%
}
\newcommand*{\@do@write@glslabels}{%
    \AtEndDocument{\@do@write@glslabels}%
    \let\@do@write@glslabels\relax
}
\newcommand*{\@do@write@glslabels}{%
    \newwrite\@glsl@labelsfile
    \immediate\openout\@glsl@labelsfile=\jobname.glslabels
    \forallglslentries[\@gls@types,\@ignored@glossaries]{\@glslentry}%
    {\ifdefempty{\@glslentry}{\immediate\write\@glsl@labelsfile{\@glslentry}}}%
    \immediate\closeout\@glsl@labelsfile
}
\@gls@declareoption{writeglslabels}{\@do@write@glslabels}
\newif\ifglslautomake
\newcommand{\glsl@automake@nr}{1}
\define@choicekey{glossaries.sty}{automake}%
    [\glsl@automake@val\glsl@automake@nr]{true,false,immediate}[true]{%
    \ifnum\glsl@automake@nr=1\relax
        \glslautomakefalse
    \else
        \glslautomaketrue
    \fi
}

```

```

\ifglssautomake
  \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
}
\glssautomakefalse
\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][\@gls@tr@set@symbols@toctitle]{%
\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@option
      \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
\ifgl Sentrycounter
    \let\@gls@doentrycounterdef\relax
\else
    \let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
\fi
\ifgl subentrycounter
    \let\@gls@dosubentrycounterdef\relax
\else
    \let\@gls@dosubentrycounterdef\@gls@define@glossarysubentrycounter
\fi
\setkeys{glossaries.sty}{#1}%
\@gls@setacrstyle
\@gls@setupshortcuts
\@gls@do@acronymsdef
\@gls@do@numbersdef
\@gls@do@symbolssdef
\@gls@do@indexdef
\@gls@doentrycounterdef
\@gls@dosubentrycounterdef
}
\ifthenelse{\equal{\glscounter}{section}}{%
{%
    \ifcsundef{chapter}{}%
    {%
        \let\@gls@old@chapter\@chapter
        \def\@chapter[#1]#2{\@gls@old@chapter[#1]}{#2}%
        \ifcsundef{hyperdef}{\hyperdef{section}{\thesection}}{}}%
    }%
}%
{}
\newcommand*{\@gls@onlypremakeg}{%
\newcommand*{\@onlypremakeg}[1]{%
    \ifx\@gls@onlypremakeg\@empty
        \def\@gls@onlypremakeg{#1}%
    \else
        \expandafter\toks@\expandafter{\@gls@onlypremakeg}%
        \edef\@gls@onlypremakeg{\the\toks@,\noexpand#1}%
    \fi
}
\newcommand*{\@disable@onlypremakeg}{%
\@for\@thiscs:=\@gls@onlypremakeg\do{%

```

```

\expandafter\@disable@premakecs\@thiscs%
}}
\newcommand*{\@disable@premakecs}[1]{%
  \def#1{\PackageError{glossaries}{\string#1\space may only be
    used before \string\makeglossaries}{You can't use
    \string#1\space after \string\makeglossaries}}}%
}
\providecommand*{\glossaryname}{Glossary}
\providecommand*{\acronymname}{Acronyms}
\newcommand*{\glssettoctitle}[1]{%
  \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}
\providecommand*{\entryname}{Notation}
\providecommand*{\descriptionname}{Description}
\providecommand*{\symbolname}{Symbol}
\providecommand*{\pagelistname}{Page List}
\providecommand*{\glsymbolsgroupname}{Symbols}
\providecommand*{\glsnumbersgroupname}{Numbers}
\newcommand*{\glspluralsuffix}{s}
\newcommand*{\glsacrpluralsuffix}{\glspluralsuffix}
\newcommand*{\glsupacrpluralsuffix}{\glsup{t}\glsacrpluralsuffix}
\providecommand*{\seename}{see}
\providecommand*{\andname}{\&}
\newcommand*{\RequireGlossariesLang}[1]{%
  \ifundefined{ver@glossaries-#1.1df}{\input{glossaries-#1.1df}}{}}%
}
\newcommand*{\ProvidesGlossariesLang}[1]{%
  \ProvidesFile{glossaries-#1.1df}%
}
\newcommand*{\addglossarytocaptions}[1]{}
\ifglstranslate
  \RequirePackage{tracklang}
  \@gls@usetranslator
  \ifpackageloaded{translator}
  {%
    \ifboolexpr
    {
      test {\ifdefstring{\trans@languages}{English}}
      and not
      test {\ifdefstring{bbl@loaded}{english}}
    }
    {%
      \let\glsifusetranslator\@secondoftwo
    }%
    {%
      \usedictionary{glossaries-dictionary}%
      \renewcommand*{\addglossarytocaptions}[1]{%
        \ifcsundef{captions#1}{}%
        {%
          \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
          \expandafter\toks@\expandafter{\@gls@tmp

```

```

        \renewcommand*{\glossaryname}{\translate{Glossary}}}%
    }%
    \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
  }%
}%
}%
}%
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
  }%
}%
}%
}%
\glsifusetranslator
{%
  \renewcommand*{\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*{\glossaryname}{\translate{Glossary}}}%
  \renewcommand*{\acronymname}{\translate{Acronyms}}}%
  \renewcommand*{\entryname}{\translate{Notation (glossaries)}}}%
  \renewcommand*{\descriptionname}{%
    \translate{Description (glossaries)}}%
  \renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}}%
  \renewcommand*{\pagelistname}{%
    \translate{Page List (glossaries)}}%
  \renewcommand*{\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*{\gl@istfilebase}{#1}%
  \ifglxsindy
    \def\istfilename{\gl@istfilebase.xdy}
  \else
    \def\istfilename{\gl@istfilebase.ist}
  \fi
}
\@onlypremakeg\setStyleFile
\ifglxsindy
  \def\istfilename{\gl@istfilebase.xdy}
\else
  \def\istfilename{\gl@istfilebase.ist}
\fi
\newcommand*{\gl@istfilebase}{\jobname}
\newcommand*{\@istfilename}[1]{%
\newcommand*{\glscpositor}{.}
\newcommand*{\glSetCompositor}[1]{%
  \renewcommand*{\glscpositor}{#1}}
\@onlypremakeg\glSetCompositor
\newcommand*{\@gl@Alphacompositor}{\glscpositor}
\ifglxsindy
  \newcommand*{\glSetAlphaCompositor}[1]{%
    \renewcommand*{\@gl@Alphacompositor}{#1}}
\else
  \newcommand*{\glSetAlphaCompositor}[1]{%
    \gl@noxindywarning\glSetAlphaCompositor}
\fi
\@onlypremakeg\glSetAlphaCompositor
\newcommand*{\gl@suffixF}{%
\newcommand*{\glSetSuffixF}[1]{%
  \renewcommand*{\gl@suffixF}{#1}}
\@onlypremakeg\glSetSuffixF
\newcommand*{\gl@suffixFF}{%
\newcommand*{\glSetSuffixFF}[1]{%
  \renewcommand*{\gl@suffixFF}{#1}}
}
\ifcsundef{hyperlink}%
{%
  \newcommand*{\gl@numberformat}[1]{#1}%
}%
{%
  \newcommand*{\gl@numberformat}[1]{\glshypernumber{#1}}%
}

```

```

\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*{\glossarypreamble}{%
  \csuse{@glossarypreamble@\currentglossary}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaultttype]{%
  \ifglossaryexists*{#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*{\glossarypostamble}{}
\newcommand*{\glossarysection}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@p@glossarysection{#1}{#2}%
  }%
  \gls@glossarymark{\glossarytoctitle}%
}
\ifcsundef{glossarymark}%
{%
  \newcommand{\gls@glossarymark}[1]{\glossarymark{#1}}
}%
{%
  \ifclassloaded{memoir}
  {%
    \newcommand{\gls@glossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUHead{#1}}{\memUHead{#1}}%
      \else
        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand{\gls@glossarymark}[1]{%
      \ifglsucmark
        \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
      \else
        \@mkboth{#1}{#1}%
      \fi
    }
  }
}
\providecommand{\glossarymark}[1]{%
  \ifglsucmark
    \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%

```

```

\else
  \@mkboth{#1}{#1}%
\fi
}
\newcommand*\setglossarysection[1]{%
\setkeys{glossaries.sty}{section=#1}}
\newcommand*\@glossarysection[2]{%
\ifdefempty\@glossarysecstar
{%
\csname\@glossarysec\endcsname[#1]{#2}%
}%
{%
\csname\@glossarysec\endcsname*{#2}%
\@gls@toc{#1}{\@glossarysec}%
}%
\@glossaryseclabel
}
\newcommand*\@pglossarysection[2]{%
\gls@clearpage
\phantomsection
\ifdefempty\@glossarysecstar
{%
\csname\@glossarysec\endcsname{#2}%
}%
{%
\@gls@toc{#1}{\@glossarysec}%
\csname\@glossarysec\endcsname*{#2}%
}%
\@glossaryseclabel
}
\newcommand*\gls@doclearpage{%
\ifthenelse{\equal{\@glossarysec}{chapter}}{%
{%
\ifcsundef{cleardoublepage}%
{%
\clearpage
}%
{%
\ifcsdef{if@openright}%
{%
\if@openright
\cleardoublepage
\else
\clearpage
\fi
}%
{%
\cleardoublepage
}%
}%
}

```

```

}%
{}%
}
\newcommand*{\glsclearpage}{\gls@doclearpage}
\newcommand*{\@gls@toc}[2]{%
  \ifglstoc
    \ifglsnumberline
      \addcontentsline{toc}{#2}{\protect\numberline{#1}}%
    \else
      \addcontentsline{toc}{#2}{#1}%
    \fi
  \fi
}
\newcommand*{\glsnoindywarning}[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]{%
    \glsnxindywarning\GlsAddXdyAttribute
}
\fi
\newcommand*\@disabled@glssaddxdycounters{%
    \PackageError{glossaries}{\string\GlsAddXdyCounters\space
can't be used after \string\GlsAddXdyAttribute}{Move all
occurrences of \string\GlsAddXdyCounters\space before the first
instance of \string\GlsAddXdyAttribute}%
}
\ifglsxindy
\newcommand*\@glssaddxdyattribute[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\@glssaddxdyattribute{#1}{\@this@counter}%
        }
        \gls@do@addxdyattribute
    }%
    \let\GlsAddXdyCounters\@disabled@glssaddxdycounters
}
\@onlypremakeg\GlsAddXdyAttribute
\else
\newcommand*\GlsAddXdyAttribute[1]{%
    \glsnxindywarning\GlsAddXdyAttribute}
\fi

```

```

\ifglxsindy
\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{}
\ifglxsindy
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \edef\@xdyuseralphabets{%
      \@xdyuseralphabets ^~J
      (define-alphabet "#1" (#2))}%
  }
\else
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \glsnoxywarning\GlsAddXdyAlphabet}
\fi
\ifglxsindy
  \newcommand*{\@gls@xdy@locationlist}{%
    roman-page-numbers,%
    Roman-page-numbers,%
    arabic-page-numbers,%
    alpha-page-numbers,%
    Alpha-page-numbers,%
    Appendix-page-numbers,%
    arabic-section-numbers%
  }
\protected@edef\@gls@roman{\@roman{0}\string"

```

```

\string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@gls@roman
\edef\@tmp{\string" \string"roman-numbers-lowercase\string"
:sep \string"}%
\@onelevel@sanitize\@tmp
\ifx\@tmp\@gls@roman
\expandafter
\edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
\string"roman-numbers-lowercase\string"%
}%
\else
\expandafter
\edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{
:sep \string"\@gls@roman\string"%
}%
\fi
\expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
\string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
\string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
\string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
\string"ALPHA\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
\string"ALPHA\string"
:sep \string"\@gls@Alphacompositor\string"
\string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
\string"arabic-numbers\string"
:sep \string"\glscompositor\string"
\string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglxindy
\newcommand\@xdycrossrefhook{}
\fi
\ifglxindy
\newcommand*\@GlsAddXdyLocation}[3][[]]{%
\def\@gls@tmp{#1}%
\ifx\@gls@tmp\@empty
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%

```

```

        (define-location-class \string"#2\string"^^J\space\space
        \space(:sep \string"{}\glsoopenbrace\string" #3
        :sep \string"\glsclosebrace\string"))
    }%
\else
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep "\glsoopenbrace"
#1
:sep "\glsclosebrace\glsoopenbrace" #3
:sep "\glsclosebrace"))
}%
\fi
\edef\@xdyuserlocationnames{%
\@xdyuserlocationnames^^J\space\space\space
\string"#2\string"}%
}
\@onlypremakeg\GlsAddXdyLocation
\else
\newcommand*\GlsAddXdyLocation}[2]{%
\glsnnoxindywarning\GlsAddXdyLocation}
\fi
\ifglxindy
\def\@xdylocationclassorder{^^J\space\space\space
\string"roman-page-numbers\string"^^J\space\space\space
\string"arabic-page-numbers\string"^^J\space\space\space
\string"arabic-section-numbers\string"^^J\space\space\space
\string"alpha-page-numbers\string"^^J\space\space\space
\string"Roman-page-numbers\string"^^J\space\space\space
\string"Alpha-page-numbers\string"^^J\space\space\space
\string"Appendix-page-numbers\string"
\@xdyuserlocationnames^^J\space\space\space
\string"see\string"
}
\fi
\ifglxindy
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\def\@xdylocationclassorder{#1}}
\else
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\glsnnoxindywarning\GlsSetXdyLocationClassOrder}
\fi
\ifglxindy
\def\@xdysortrules{}
\fi
\ifglxindy
\newcommand*\GlsAddSortRule[2]{%
\expandafter\toks@\expandafter{\@xdysortrules}%
\protected@edef\@xdysortrules{\the\toks@ ^^J

```

```

        (sort-rule \string"#1\string" \string"#2\string"))}%
    }
\else
    \newcommand*\GlsAddSortRule[2]{%
        \glsnoxywarning\GlsAddSortRule}
\fi
\ifglxindy
    \def\@xdyrequiredstyles{tex}
\fi
\ifglxindy
    \newcommand*\GlsAddXdyStyle[1]{%
        \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
    \newcommand*\GlsAddXdyStyle[1]{%
        \glsnoxywarning\GlsAddXdyStyle}
\fi
\ifglxindy
    \newcommand*\GlsSetXdyStyles[1]{%
        \edef\@xdyrequiredstyles{#1}}
\else
    \newcommand*\GlsSetXdyStyles[1]{%
        \glsnoxywarning\GlsSetXdyStyles}
\fi
\newcommand*\findrootlanguage{}{}
\def\@xdylanguage#1#2{}
\ifglxindy
    \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
        \ifglossaryexists{#1}{%
            \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
        }{%
            \PackageError{glossaries}{Can't set language type for
                glossary type '#1' --- no such glossary}{%
                You have specified a glossary type that doesn't exist}}}%
\else
    \newcommand*\GlsSetXdyLanguage[2][{}]{%
        \glsnoxywarning\GlsSetXdyLanguage}
\fi
\def\@gls@codepage#1#2{}
\ifglxindy
    \newcommand*\GlsSetXdyCodePage[1]{%
        \renewcommand*\@gls@codepage{#1}%
    }
    \AtBeginDocument{%
        \ifx\@gls@codepage\@empty
            \ifpackageloaded{fontspec}{\def\@gls@codepage{utf8}}{}%
        \fi
    }
\else
    \newcommand*\GlsSetXdyCodePage[1]{%
        \glsnoxywarning\GlsSetXdyCodePage}

```

```

\fi
\ifglxsindy
\ifglxsindy@glslnumbers
\def\@xdylettergroups{(define-letter-group
\string"glslnumbers\string"^^J\space\space\space
:prefixes (\string"0\string" \string"1\string"
\string"2\string" \string"3\string" \string"4\string"
\string"5\string" \string"6\string" \string"7\string"
\string"8\string" \string"9\string")^^J\space\space\space
\@xdynumbergrouporder)}
\else
\def\@xdylettergroups{}
\fi
\fi
\newcommand*\GlsAddLetterGroup[2]{%
\expandafter\toks@\expandafter{\@xdylettergroups}%
\protected@edef\@xdylettergroups{\the\toks@^^J%
(define-letter-group \string"#1\string"^^J\space\space\space#2)}}%
}%
\newcommand*\forallglossaries[3][\@glo@types]{%
\@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%
}

\newcommand*\forallacronyms[2]{%
\@for#1:=\@glslacronymlists\do{\ifx#1\@empty\else#2\fi}%
}
\newcommand*\forallglsentries[3][\@glsldefaulttype]{%
\edef\@glo@list{\csname glolist@#1\endcsname}%
\@for#2:=\@glo@list\do
{%
\ifdefempty{#2}{\fi}%
}%
}
\newcommand*\forallglsentries[3][\@glo@types]{%
\expandafter\forallglossaries\expandafter[#1]{\@this@glo@}%
{%
\forallglsentries[\@this@glo@]{#2}{#3}%
}%
}
\newcommand*\ifglossaryexists{%
\@ifstar\s@ifglossaryexists\@ifglossaryexists
}
\newcommand*\@ifglossaryexists[3]{%
\ifcsundef{\@glo@type@#1@out}{#3}{#2}%
}
\newcommand*\s@ifglossaryexists[3]{%
\ifcsundef{glolist@#1}{#3}{#2}%
}
\newcommand*\glslsetoklabel[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\for@glseentries\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]{%
    \glstoifexists{#1}%
    {%
        \ifcsepty{glo@\glstetoklabel{#1}@parent}{#3}{#2}%
    }%
}
\newcommand*{\ifglshasdesc}[3]{%
    \ifcsepty{glo@\glstetoklabel{#1}@desc}%
    {#3}%
    {#2}%
}
\newcommand*{\ifglshasdescsuppressed}[3]{%
    \ifcsequal{glo@\glstetoklabel{#1}@desc}{@no@post@desc}%
    {#2}%
    {#3}%
}
}
\newrobustcmd*{\ifglshassymbol}[3]{%
    \letcs{\@glo@symbol}{glo@\glstetoklabel{#1}@symbol}%
    \ifdefempty\@glo@symbol
    {#3}%
    {%
        \ifdefequal\@glo@symbol\@gls@default@value
        {#3}%
        {#2}%
    }%
}
}
\newrobustcmd*{\ifglshaslong}[3]{%
    \letcs{\@glo@long}{glo@\glstetoklabel{#1}@long}%
    \ifdefempty\@glo@long
    {#3}%
    {%
        \ifdefequal\@glo@long\@gls@default@value
        {#3}%
        {#2}%
    }%
}
}
\newrobustcmd*{\ifglshasshort}[3]{%
    \letcs{\@glo@short}{glo@\glstetoklabel{#1}@short}%
    \ifdefempty\@glo@short
    {#3}%
    {%
        \ifdefequal\@glo@short\@gls@default@value

```



```

        {#3}%
        {#2}%
    }%
}
\newrobustcmd*{\ifglshasfield}[4]{%
    \glsdoifexists{#2}%
    {%
        \letcs{\@glo@thisvalue}{glo@glstetoklabel{#2}@#1}%
        \ifdef\@glo@thisvalue
        {%
            \ifdefempty\@glo@thisvalue
            {%
                #4%
            }%
        }%
        {%
            \ifdefequal\@glo@thisvalue\@gls@default@value
            {%
                #4%
            }%
            {%
                \let\glscurrentfieldvalue\@glo@thisvalue
                #3%
            }%
        }%
    }%
}%
{%
    \@gls@fetchfield{\@gls@thisfield}{#1}%
    \ifdef\@gls@thisfield
    {%
        \letcs{\@glo@thisvalue}{glo@glstetoklabel{#2}@ \@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][\glsdefaultttype]{%
  \csgdef{gls@#1@entryfmt}{#2}%
}%
\newcommand*{\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*{\@gls@forbidtextext}[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\glsdefaultttype
    {%
      \gdef\glsdefaultttype{#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@forbidtextext\csname @glotype@#2@log\endcsname
  \expandafter\@gls@forbidtextext\csname @glotype@#2@in\endcsname
  \expandafter\@gls@forbidtextext\csname @glotype@#2@out\endcsname
  \expandafter\def\csname @glotype@#2@title\endcsname{#5}%
  \@gls@provide@newglossary
  \protected@write\@auxout{}\string\@newglossary{#2}{#1}{#3}{#4}%
  \ifcsundef{gls@#2@entryfmt}%

```

```

{%
  \defglsentryfmt[#2]{\glsentryfmt}%
}%
{}%
\@gls@defsortcount{#2}%
\@ifnextchar[{\@gls@setcounter{#2}}%
  {\@gls@setcounter{#2}[\glscounter]}%
}%
{%
  \gls@gobbleopt
}%
}
\newcommand*{\altnewglossary}[3]{%
  \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
}
\@onlypreamble{\newglossary}
\@onlypremakeg\newglossary
\newcommand*{\@newglossary}[4]{}
\def\@gls@setcounter#1[#2]{%
  \expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
  \ifglxsindy
    \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*{\@glsnodelsc{%
  \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}%
      }%
    }%
  }%
}
\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]{%
  \ifglentryexists{#1}%
  {}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\@onlypreamble{\provideglossaryentry}
\newrobustcmd{\new@glossaryentry}[2]{%
  \ifundef\@gls@deffile
  {%
    \global\newwrite\@gls@deffile
    \immediate\openout\@gls@deffile=\jobname.glsdefs
  }%
  {}%
  \ifglentryexists{#1}{}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
  \@gls@writedef{#1}%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*{\gls@begindocdefs}{%
  \@gls@enablesavenonumberlist
  \edef\@gls@restoreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
  \makeatletter
  \InputIfFileExists{\jobname.glsdefs}{\relax}{\relax}%
  \@gls@restoreat
  \undef\@gls@restoreat
  \gls@defdocnewglossaryentry
}
\newcommand*{\@gls@writedef}[1]{%
  \immediate\write\@gls@deffile
  {%
    \string\ifglentryexists{#1}{\glspercentchar^^J%
    \expandafter\@gobble\string{\glspercentchar^^J%
      \string\gls@defglossaryentry{\glsdetoklabel{#1}}\glspercentchar^^J%
      \expandafter\@gobble\string{\glspercentchar%
    }%
    \@for\@gls@map:=\@gls@keymap\do
    {%
      \letcs\glo@value{glo@\glsdetoklabel{#1}}\expandafter\@secondoftwo\@gls@map}%
      \ifdef\glo@value
      {%
        \@onelevel@sanitize\glo@value
        \immediate\write\@gls@deffile
        {%
          \expandafter\@firstoftwo\@gls@map
          =\expandafter\@gobble\string{\glo@value\expandafter\@gobble\string\},%
        }%
      }%
    }%
  }%
}

```

```

        \glspercentchar
    }%
}%
{}%
}%
\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{\glsfieldxddef}[3]{%
    \glsdoifexists{#1}%
    {%
        \edef\@glo@label{\glsdetoklabel{#1}}%
        \ifcsdef{glo@\@glo@label @#2}%
        {%
            \protected@csxdef{glo@\@glo@label @#2}{#3}%
        }%
        {%
            \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
        }%
    }%
}
\newcommand{\glsfielddedef}[3]{%
    \glsdoifexists{#1}%
    {%
        \edef\@glo@label{\glsdetoklabel{#1}}%
        \ifcsdef{glo@\@glo@label @#2}%
        {%
            \protected@csedef{glo@\@glo@label @#2}{#3}%
        }%
        {%
            \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
        }%
    }%
}
\newcommand{\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\@gls@name
\let\@glo@desc\@gls@desc
\let\@glo@descplural\@gls@descplural
\let\@glo@type\@gls@type
\let\@glo@symbol\@gls@symbol
\let\@glo@symbolplural\@gls@symbolplural
\let\@glo@text\@gls@text
\let\@glo@plural\@gls@plural
\let\@glo@first\@gls@first
\let\@glo@firstplural\@gls@firstplural
\let\@glo@sort\@gls@sort
\let\@glo@counter\@gls@counter
\def\@glo@see{ }%
\def\@glo@parent{ }%
\def\@glo@prefix{ }%
\@gls@initnonnumberlist
\def\@glo@useri{ }%
\def\@glo@userii{ }%
\def\@glo@useriii{ }%
\def\@glo@useriv{ }%
\def\@glo@userv{ }%
\def\@glo@uservi{ }%
\def\@glo@short{ }%
\def\@glo@shortpl{ }%
\def\@glo@long{ }%
\def\@glo@longpl{ }%
\@newglossaryentryprehook
\setkeys{glossentry}{#2}%
\ifundef\glsdefaulttype
{%
  \PackageError{glossaries}%
  {No default glossary type (have you used ‘nomain’ by mistake?)}%
  {If you use package option ‘nomain’ you must define
   a new glossary before you can define entries}%
}%
{}%
\gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
\edef\@glo@type{\glsentrytype{\@glo@label}}%
\ifcsundef{glo@list@\@glo@type}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘\@glo@type’ has not been defined}%
  {You need to define a new glossary type, before making entries
   in it}%
}%
{%
  \ifignoredglossary\@glo@type
  {%
    \ifx\@glo@desc\@gls@desc

```



```

        \let\@glo@desc\@empty
    \fi
}%
{%
}%
\protected@edef\@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\@glo@name\@gls@default@value
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%
\ifcsundef{glo@\@glo@label @numberlist}%
{%
  \csxdef{glo@\@glo@label @numberlist}{%
    \noexpand\@gls@missingnumberlist{\@glo@label}}%
}%
{%
  \@gls@storenonumberlist{\@glo@label}%
  \def\@glo@@desc{\@glo@first}%
  \ifx\@glo@desc\@glo@@desc

```

```

\let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@glsnodesc
\@glsnodesc
\let\@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}{%
\ifdefined\@glo@see{%
{%
\protected@edef\@do@glssee{%
\noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
\noexpand\expandafter\noexpand\@glssee\noexpand\@glo@list{\@glo@label}}%
\@do@glssee
}%
\@glo@autoseehook
}%
\newcommand*{\@glo@autoseehook}{%

```

```

\newcommand*{\@newglossaryentryprehook}{}
\newcommand*{\@newglossaryentryposthook}{}
\newcommand*{\@newglossaryentry@defcounters}{}
\newcommand*{\glsmoveentry}[2]{%
  \edef\@glo@thislabel{\glsdetoklabel{#1}}%
  \edef\@glo@type{\csname glo@\@glo@thislabel @type\endcsname}%
  \def\@glo@list{,%}
  \for@gl@entries[\@glo@type]{\@glo@label}%
  {%
    \if@defequal\@glo@thislabel\@glo@label
      {}{\@eappto\@glo@list{\@glo@label},}%
    }%
  \cslet\@glolist\@glo@type{\@glo@list}%
  \csdef{glo@\@glo@thislabel @type}{#2}%
}
\ifgl@xindy
  \newcommand*{\@glossaryentryfield}{\string\@glossentry}
\else
  \newcommand*{\@glossaryentryfield}{\string\glossentry}
\fi
\ifgl@xindy
  \newcommand*{\@glossarysubentryfield}{%
    \string\@subglossentry}
\else
  \newcommand*{\@glossarysubentryfield}{%
    \string\subglossentry}
\fi
\newcommand{\@glo@storeentry}[1]{%
  \edef\@glo@esclabel{#1}%
  \@gls@checkmkidxchars\@glo@esclabel
  \protected@edef\@glo@sort{\csname glo@#1@sort\endcsname}%
  \@gls@checkmkidxchars\@glo@sort
  \@gls@checkmkidxchars\@glo@prefix
  \edef\@glo@parent{\csname glo@#1@parent\endcsname}%
  \ifgl@xindy
    \ifx\@glo@parent\@empty
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        (\string"\@glo@sort\string" %
        \string"\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %
      }%
    \else
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \csname glo@\@glo@parent @index\endcsname
        (\string"\@glo@sort\string" %
        \string"\@glo@prefix\@glossarysubentryfield
        {\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
      }%
    \fi
  \else
    \ifx\@glo@parent\@empty

```

```

\@onelevel@sanitize\@glo@prefix
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
  \@glo@sort\@gls@actualchar\@glo@prefix
  \@glossaryentryfield{\@glo@esclabel}%
}%
\else
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
  \csname glo@\@glo@parent @index\endcsname\@gls@levelchar
  \@glo@sort\@gls@actualchar\@glo@prefix
  \@glossarysubentryfield
    {\csname glo@#1@level\endcsname}{\@glo@esclabel}%
}%
\fi
\fi
}
\AtBeginDocument{%
  \@ifpackageloaded{amsmath}%
  {\let\gls@ifnotmeasuring\@gls@ifnotmeasuring}%
  {}%
}
\newcommand*{\@gls@ifnotmeasuring}[1]{%
  \ifmeasuring@
  \else
    #1%
  \fi
}
\newcommand*\gls@ifnotmeasuring[1]{#1}
\def\@gls@patchtabularx#1\hbox#2#3!!{%
  \def\TX@trial##1{#1\hbox{\let\glsunset\@gobble#2}#3}%
}
\newcommand*\glspatchtabularx{%
  \ifdef\TX@trial
  {%
    \expandafter\@gls@patchtabularx\TX@trial{##1}!!%
    \let\glspatchtabularx\relax
  }%
  {}%
}
\newcommand*{\glsreset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glsreset{#1}%
    }%
  }%
}
\newcommand*{\glslocalreset}[1]{%
  \gls@ifnotmeasuring
  {%

```

```

\glsdoifexists{#1}%
{%
    \@glslocalreset{#1}%
}%
}%
}
\newcommand*{\glsunset}[1]{%
    \gls@ifnotmeasuring
    {%
        \glsdoifexists{#1}%
        {%
            \@glsunset{#1}%
        }%
    }%
}%
}
\newcommand*{\glslocalunset}[1]{%
    \gls@ifnotmeasuring
    {%
        \glsdoifexists{#1}%
        {%
            \@glslocalunset{#1}%
        }%
    }%
}%
}
\newcommand*{\@glslocalunset}{\@glslocalunset}
\newcommand*{\@glslocalunset}[1]{%
    \expandafter\let\csname ifglo@glstetoklabel{#1}@flag\endcsname\iftrue
}%
\newcommand*{\@glsunset}{\@glsunset}
\newcommand*{\@glsunset}[1]{%
    \expandafter\global\csname glo@glstetoklabel{#1}@flagtrue\endcsname
}%
\newcommand*{\@glslocalreset}{\@glslocalreset}
\newcommand*{\@glslocalreset}[1]{%
    \expandafter\let\csname ifglo@glstetoklabel{#1}@flag\endcsname\iffalse
}%
\newcommand*{\@glsreset}{\@glsreset}
\newcommand*{\@glsreset}[1]{%
    \expandafter\global\csname glo@glstetoklabel{#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]{%

```



```

\newcommand*{\@gls@write@entrycounts}{%
  \immediate\write\@auxout
  {\string\providecommand*\string\@gls@entry@count}[2]{}}%
\forallglsentries{\@glsentry}{%
  \ifglsused{\@glsentry}%
  {\immediate\write\@auxout
    {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}%
  }%
}%
}
\newcommand*{\@gls@entry@count}[2]{%
\newrobustcmd*{\cgl}{\@gls@hyp@opt\@cgl}
\newcommand*{\@cgl}[2][{}]{%
  \new@ifnextchar[{\@cgl@{#1}{#2}}{\@cgl@{#1}{#2}}[{}]}%
}
\def\@cgl@#1#2[#3]{%
  \GlossariesWarning{\string\cgl\space is defaulting to
    \string\gls\space since you haven't enabled entry counting}%
  \@gls@{#1}{#2}[#3]%
}
\newcommand*{\cglformat}[2]{%
  \ifglshaslong{#1}{\glsentrylong{#1}}{\glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cGls}{\@gls@hyp@opt\@cGls}
\newcommand*{\@cGls}[2][{}]{%
  \new@ifnextchar[{\@cGls@{#1}{#2}}{\@cGls@{#1}{#2}}[{}]}%
}
\def\@cGls@#1#2[#3]{%
  \GlossariesWarning{\string\cGls\space is defaulting to
    \string\Gls\space since you haven't enabled entry counting}%
  \@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}%
        }%
    }%
}%

```

```

{%
  \ifglssused\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}%
    }%
  }%
  {%
    \ifglssused\glslabel
    {%
      \mfirstucMakeUppercase{#2{\glsentrytext{\glslabel}}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{\glsinsert}}%
    }%
    {%
      \mfirstucMakeUppercase{#1{\glsentryfirst{\glslabel}}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{\glsinsert}}%
    }%
  }%
  {%
    \ifglssused{\glslabel}%

```

```

    {%
      #2{\glscustomtext}%
      {\glentrydesc{\glslabel}}%
      {\glentrysymbol{\glslabel}}{%}%
    }%
    {%
      #1{\glscustomtext}%
      {\glentrydesc{\glslabel}}%
      {\glentrysymbol{\glslabel}}{%}%
    }%
  }%
}
\newcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glusifplural
    {%
      \glscapscase
      {%
        \ifglused\glslabel
        {%
          \glentryplural{\glslabel}\glsinsert
        }%
        {%
          \glentryfirstplural{\glslabel}\glsinsert
        }%
      }%
    }%
    {%
      \ifglused\glslabel
      {%
        \Glsentryplural{\glslabel}\glsinsert
      }%
      {%
        \Glsentryfirstplural{\glslabel}\glsinsert
      }%
    }%
    {%
      \ifglused\glslabel
      {%
        \mfirstucMakeUppercase
        {\glentryplural{\glslabel}\glsinsert}%
      }%
      {%
        \mfirstucMakeUppercase
        {\glentryfirstplural{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glscapscase

```

```

{%
  \ifglused\glslabel
  {%
    \glsentrytext{\glslabel}\glsinsert
  }%
  {%
    \glsentryfirst{\glslabel}\glsinsert
  }%
}%
{%
  \ifglused\glslabel
  {%
    \Glsentrytext{\glslabel}\glsinsert
  }%
  {%
    \Glsentryfirst{\glslabel}\glsinsert
  }%
}%
{%
  \ifglused\glslabel
  {%
    \mfirstucMakeUppercase{\glsentrytext{\glslabel}\glsinsert}%
  }%
  {%
    \mfirstucMakeUppercase{\glsentryfirst{\glslabel}\glsinsert}%
  }%
}%
}%
{%
  \glscustomtext\glsinsert
}%
}
\newcommand*{\glsgenacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglused\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]{%
  \glstrylong{#1}#2\space
  (\protect\firstacronymfont{\glstryshort{#1}})}%
}
\newcommand*{\Genacrfullformat}[2]{%
  \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
  \xmakefirstuc\gls@text
}
\newcommand*{\genplacrfullformat}[2]{%
  \glstrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glstryshortpl{#1}})}%
}
\newcommand*{\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \xmakefirstuc\gls@text
}
\newcommand*{\glsdisplayfirst}[4]{#1#4}
\newcommand*{\glsdisplay}[4]{#1#4}
\newcommand*{\defglsdisplay}[2][\glsdefaulttype]{%
  \GlossariesWarning{\string\defglsdisplay\space is now obsolete.^^J
  Use \string\defglsentryfmt\space instead}%
  \expandafter\def\csname gls@#1@display\endcsname##1##2##3##4{#2}%
  \edef\@gls@doentrydef{%
    \noexpand\defglsentryfmt[#1]{%
      \noexpand\ifcsdef{gls@#1@displayfirst}%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\glsdisplayfirst}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
    }%
  }%
  \@gls@doentrydef
}
\newcommand*{\defglsdisplayfirst}[2][\glsdefaulttype]{%
  \GlossariesWarning{\string\defglsdisplayfirst\space is now obsolete.^^J
  Use \string\defglsentryfmt\space instead}%
  \expandafter\def\csname gls@#1@displayfirst\endcsname##1##2##3##4{#2}%
  \edef\@gls@doentrydef{%
    \noexpand\defglsentryfmt[#1]{%
      \noexpand\ifcsdef{gls@#1@display}%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
    }%
  }%
  \@gls@doentrydef
}

```



```

    }%
    {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\glsdisplay}%
    }%
    }%
    }%
    \@gls@doentrydef
}
\define@key{glslink}{counter}{%
    \ifcsundef{c@#1}%
    {%
        \PackageError{glossaries}%
        {There is no counter called '#1'}%
        {%
            The counter key should have the name of a valid counter
            as its value%
        }%
    }%
    }%
    {%
        \def\@gls@counter{#1}%
    }%
}
\define@key{glslink}{format}{%
    \def\@glsnumberformat{#1}}
\define@boolkey{glslink}{hyper}[true]{}
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\define@boolkey{glslink}{local}[true]{}
\newcommand*{\glslinkvar}[3]{#1}
\newcommand*{\glsifhyper}[2]{%
    \glslinkvar{#1}{#2}{#1}%
    \GlossariesWarning{\string\glsifhyper\space is deprecated. Did
        you mean \string\glsifhyperon\space or \string\glslinkvar?}%
}
\newcommand*{\@gls@hyp@opt}[1]{%
    \let\glslinkvar\@firstofthree
    \let\@gls@hyp@opt@cs#1\relax
    \@ifstar{\s@gls@hyp@opt}%
    {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*{\s@gls@hyp@opt}[1][]{%
    \let\glslinkvar\@secondofthree
    \@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1][]{%
    \let\glslinkvar\@thirdofthree
    \@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
    \@gls@hyp@opt\@gls@link
}

```

```

\newcommand*{\@gls@link}[3][\{%
  \glsdoifexistsordo{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \@gls@link[#1]{#2}{#3}%
  }{%
    \glstextformat{#3}%
  }%
  \glspostlinkhook
}
\newcommand*{\glspostlinkhook}{}
\newcommand*{\@gls@link@checkfirsthyper}{%
  \ifglsused{\glslabel}%
  {%
  }%
  {%
    \gls@checkisacronymlist\glstype
    \ifglshyperfirst
      \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\theHglentrycounter{\@gls@Hcounter}%
          \restorecounters@
        \fi
      }%
    }%
    {%
      \edef\theglsentrycounter{\expandafter\noexpand
        \csname the\@gls@counter\endcsname}%
    }%
    \ifx\@gls@Hcounter\@empty
      \ifcsundef{theH\@gls@counter}%
      {%
        \def\theHglentrycounter{\theglsentrycounter}%
      }%
    }%
  }%
}

```

```

}%
{%
\protected@edef\theHglentrycounter{\expandafter\noexpand
\csname theH\@gls@counter\endcsname}%
}%
\fi
}
\def\@set@glo@numformat#1#2#3#4{%
\expandafter\@glo@check@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@dostsubst{%
  \noexpand\DTLsubstituteall\noexpand\gls@xdystring
  {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
}%
\gls@dostsubst
}%
\let#1=\gls@xdystring
}
\newcommand{\@gls@checkmkidxchars}[1]{%
  \ifglsxindy
    \gls@escbsdq{#1}%
  \else
    \def\@gls@checkedmkidx{%
      \expandafter\@gls@checkquote#1\@nil""\null
      \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
      \def\@gls@checkedmkidx{%
        \expandafter\@gls@checkescquote#1\@nil""\null
        \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
        \def\@gls@checkedmkidx{%
          \expandafter\@gls@checkescactual#1\@nil"??\null
          \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
          \def\@gls@checkedmkidx{%
            \expandafter\@gls@checkactual#1\@nil??\null
            \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
            \def\@gls@checkedmkidx{%
              \expandafter\@gls@checkbar#1\@nil||\null
              \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
              \def\@gls@checkedmkidx{%
                \expandafter\@gls@checkescbar#1\@nil|||\null
                \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                \def\@gls@checkedmkidx{%
                  \expandafter\@gls@checklevel#1\@nil!!\null
                  \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                }
              }
            }
          }
        }
      }
    }
  \fi
}
\def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
\newtoks\@gls@tmpb
\def\@gls@checkquote#1"#2"#3\null{%
  \gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\@nil#2\@nil
    \ifx\@nil#3\@nil
      \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\@nil}%
    \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{%

```

```

\newcommand\def\newcommand\@gls@xdycheckbackslash##1\@backslashchar
##2\@backslashchar##3\newcommand\null{%
\newcommand\@gls@tmpb=\newcommand\expandafter
{\newcommand\@gls@checkedmkidx}%
\newcommand\toks@={##1}%
\newcommand\ifx\newcommand\null##2\newcommand\null
\newcommand\ifx\newcommand\null##3\newcommand\null
\newcommand\edef\newcommand\@gls@checkedmkidx{%
\newcommand\the\newcommand\@gls@tmpb\newcommand\the\newcommand\toks@}%
\newcommand\def\newcommand\@gls@xdycheckbackslash{\relax}%
\newcommand\else
\newcommand\edef\newcommand\@gls@checkedmkidx{%
\newcommand\the\newcommand\@gls@tmpb\newcommand\the\newcommand\toks@
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
\newcommand\def\newcommand\@gls@xdycheckbackslash{%
\newcommand\@gls@xdycheckbackslash##3\newcommand\null}%
\newcommand\fi
\newcommand\else
\newcommand\edef\newcommand\@gls@checkedmkidx{%
\newcommand\the\newcommand\@gls@tmpb\newcommand\the\newcommand\toks@
\@backslashchar\@backslashchar}%
\newcommand\ifx\newcommand\null##3\newcommand\null
\newcommand\def\newcommand\@gls@xdycheckbackslash{%
\newcommand\@gls@xdycheckbackslash##2\@backslashchar
\@backslashchar\newcommand\null}%
\newcommand\else
\newcommand\def\newcommand\@gls@xdycheckbackslash{%
\newcommand\@gls@xdycheckbackslash##2\@backslashchar
##3\newcommand\null}%
\newcommand\fi
\newcommand\fi
\newcommand\@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@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
\let\glsifplural\@secondoftwo
\let\glscapscase\@secondofthree
\let\glscustomtext\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gl@link{#1}{#2}{\@glo@text}%
\ifKV@gl@link@local
\glslocalunset{#2}%
\else
\glsunset{#2}%
\fi
}%
\glspostlinkhook
}
\newrobustcmd*{\GLS}{\@gl@hyp@opt\@GLS}
\newcommand*{\@GLS}[2][{}]{%
\new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2}[]}%
}
\def\@GLS@#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
\let\glsifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\let\glscustomtext\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gl@link{#1}{#2}{\@glo@text}%
\ifKV@gl@link@local
\glslocalunset{#2}%
\else
\glsunset{#2}%
\fi
}%
\glspostlinkhook
}
\newrobustcmd*{\glspl}{\@gl@hyp@opt\@glspl}
\newcommand*{\@glspl}[2][{}]{%
\new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2}[]}%
}
\def\@glspl@#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
\let\glsifplural\@firstoftwo
\let\glscapscase\@firstofthree
\let\glscustomtext\@empty

```

```

\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\ifKV@glslink@local
  \glslocalunset{#2}%
\else
  \glsunset{#2}%
\fi
}%
\glspostlinkhook
}
\newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
\newcommand*{\@Glspl}[2][]{%
  \new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2}[]}%
}
\def\@Glspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\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*{\glsstext}{\@gls@hyp@opt\@glsstext}
\newcommand*{\@glsstext}[2][\@new@ifnextchar[\@glsstext@{#1}{#2}]{\@glsstext@{#1}{#2}[]}]
\def\@glsstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentrytext{#2}#3}%
}
\newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
\newcommand*{\@GLStext}[2][\@new@ifnextchar[\@GLStext@{#1}{#2}]{\@GLStext@{#1}{#2}[]}]
\def\@GLStext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrytext{#2}#3}}%
}
\newrobustcmd*{\Glstext}{\@gls@hyp@opt\@Glstext}
\newcommand*{\@Glstext}[2][\@new@ifnextchar[\@Glstext@{#1}{#2}]{\@Glstext@{#1}{#2}[]}]
\def\@Glstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentrytext{#2}#3}%
}

```

```

\newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
\newcommand*{\@glsfirst}[2][{}]{%
  \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2}[]}]
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
}
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
\newcommand*{\@Glsfirst}[2][{}]{%
  \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2}[]}]
\def\@Glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
\newcommand*{\@GLSfirst}[2][{}]{%
  \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2}[]}]
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirst{#2}#3}}%
}
\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
\newcommand*{\@glsplural}[2][{}]{%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2}[]}]
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\newcommand*{\@Glsplural}[2][{}]{%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2}[]}]
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
\newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
\newcommand*{\@GLSplural}[2][{}]{%
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2}[]}]
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryplural{#2}#3}}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{\@glsfirstplural}[2][{}]{%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2}[]}]
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
\newcommand*{\@Glsfirstplural}[2][{}]{%
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2}[]}]
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
\newcommand*{\@GLSfirstplural}[2][{}]{%

```

```

\new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} []}}
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirstplural{#2}#3}}%
}
\newrobustcmd*{\@glsname}{\@gls@hyp@opt\@glsname}
\newcommand*{\@glsname}[2] []{%
  \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2} []}}
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryname{#2}#3}%
}
\newrobustcmd*{\@GLSname}{\@gls@hyp@opt\@GLSname}
\newcommand*{\@GLSname}[2] []{%
  \new@ifnextchar[{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}}
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@GLSentryname{#2}#3}%
}
\newrobustcmd*{\@GLSname}{\@gls@hyp@opt\@GLSname}
\newcommand*{\@GLSname}[2] []{%
  \new@ifnextchar[{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}}
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryname{#2}#3}}%
}
\newrobustcmd*{\@glsdesc}{\@gls@hyp@opt\@glsdesc}
\newcommand*{\@glsdesc}[2] []{%
  \new@ifnextchar[{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2} []}}
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydesc{#2}#3}%
}
\newrobustcmd*{\@GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\newcommand*{\@GLSdesc}[2] []{%
  \new@ifnextchar[{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}}
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@GLSentrydesc{#2}#3}%
}
\newrobustcmd*{\@GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\newcommand*{\@GLSdesc}[2] []{%
  \new@ifnextchar[{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}}
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydesc{#2}#3}}%
}
\newrobustcmd*{\@glsdescplural}{\@gls@hyp@opt\@glsdescplural}
\newcommand*{\@glsdescplural}[2] []{%
  \new@ifnextchar[{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}}
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\@GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\newcommand*{\@GLSdescplural}[2] []{%
  \new@ifnextchar[{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}}
\def\@GLSdescplural@#1#2[#3]{%

```



```

\@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\newcommand*{\@GLSdescplural}[2] [] {%
  \new@ifnextchar[{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}}
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrydescplural{#2}#3}}%
}
\newrobustcmd*{\glssymbol}{\@gls@hyp@opt\@glssymbol}
\newcommand*{\@glssymbol}[2] [] {%
  \new@ifnextchar[{\@glssymbol@{#1}{#2}}{\@glssymbol@{#1}{#2} []}}
\def\@glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\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}{\mfirstucMakeUppercase{\glsentryuserv{#2}#3}}%
}
\newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
\newcommand*{\@glsuservi}[2] [] {%
  \new@ifnextchar [{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2} [] }}
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
\newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
\newcommand*{\@GLSuservi}[2] [] {%
  \new@ifnextchar [{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2} [] }}
\def\@GLSuservi@#1#2[#3]{%

```

```

\@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuservi{#2}{#3}}}%
}
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\@ns@acrshort}
\newcommand*{\ns@acrshort}[2] [] {%
  \new@ifnextchar[{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2} []}%
}
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrshort}{\@gls@hyp@opt\@ns@Acrshort}
\newcommand*{\ns@Acrshort}[2] [] {%
  \new@ifnextchar[{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2} []}%
}
\def\@Acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshort}{\@gls@hyp@opt\@ns@ACRshort}
\newcommand*{\ns@ACRshort}[2] [] {%
  \new@ifnextchar[{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}%
}
\def\@ACRshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsapscase\@thirdofthree

```

```

\let\glsinsert\@empty
\def\glscustomtext{%
  \mfirstucMakeUppercase{\acronymfont{\glsentryshort{#2}}#3}%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\acrshortpl}{\@gls@hyp@opt\@ns@acrshortpl}
\newcommand*{\ns@acrshortpl}[2][{}]{%
  \new@ifnextchar[{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2}[]}%
}
\def\@acrshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\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@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\glstryshortpl{#2}}#3}%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*\@acrlong{\@gl@hyp@opt\@ns@acrlong}
\newcommand*\@ns@acrlong[2][{}]{%
  \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2}[]}%
}
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glstrylong{#2}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*\@Acrlong{\@gl@hyp@opt\@ns@Acrlong}
\newcommand*\@ns@Acrlong[2][{}]{%
  \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2}[]}%
}
\def\@Acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylong{#2}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%

```

```

}%
\glspostlinkhook
}
\newrobustcmd*{\ACRlong}{\@gls@hyp@opt\ns@ACRlong}
\newcommand*{\ns@ACRlong}[2][]{%
  \new@ifnextchar[{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2}[]}%
}
\def\@ACRlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\glsentrylong{#2}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\ns@acrlongpl}
\newcommand*{\ns@acrlongpl}[2][]{%
  \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2}[]}%
}
\def\@acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\ns@Acrlongpl}
\newcommand*{\ns@Acrlongpl}[2][]{%
  \new@ifnextchar[{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2}[]}%
}
\def\@Acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%

```

```

\let\glsifplural\@firstoftwo
\let\glscapscase\@secondofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \Glsentrylongpl{#2}#3%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\ns@ACRlongpl}
\newcommand*{\ns@ACRlongpl}[2][{}]{%
  \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2}[]}%
}
\def\@ACRlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\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@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@name}%
    \expandafter\@gls@getbody\@glo@text{}\@nil
    \expandafter\ifx\@gls@body\glsentrylong\relax
      \expandafter\Glsentrylong\@gls@rest
    \else
      \expandafter\ifx\@gls@body\glsentryshort\relax
        \expandafter\Glsentryshort\@gls@rest
      \else
        \expandafter\ifx\@gls@body\acronymfont\relax
          {%
            \let\glsentryshort\Glsentryshort
            \@glo@text
          }%
        \else
          \expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
            {%
              \let\glsentryshort\Glsentryshort
              \@glo@text
            }%
          \else
            \xmakefirstuc{\@glo@text}%
          \fi
        \fi
      \fi
    \fi
  }%
  {%
    \@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*{\glsentryservi}[1]{%
    \@gls@entry@field{#1}{servi}%
}
\newrobustcmd*{\Glsentryservi}[1]{%
    \@Gls@entry@field{#1}{servi}%
}
\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{\glo@linkprefix\glsdetoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*{\glsadd}[2][ ]{%
  \@gls@adjustmode
  \glsdoifexists{#2}%
  {%
    \def\@glsnumberformat{\glsnumberformat}%
    \edef\@gls@counter{\csname glo@\glsdetoklabel{#2}@counter\endcsname}%
    \setkeys{glossadd}{#1}%
    \@gls@saveentrycounter
    \@gls@setsort{#2}%
    \@do@wrglossary{#2}%
  }%
}
\newcommand*{\@gls@adjustmode}{}
\AtBeginDocument{\renewcommand*{\@gls@adjustmode}{\ifvmode\mbox{}\fi}}

```

```

\newrobustcmd*{\glsaddall}[1][]{%
  \edef\@glo@type{\@glo@types}%
  \setkeys{glossadd}{#1}%
  \forallglsentries[\@glo@type]{\@glo@entry}{%
    \glsadd[#1]{\@glo@entry}%
  }%
}
\newrobustcmd*{\glsaddallunused}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glo@entry}%
  {%
    \ifglsused{\@glo@entry}{\@glsadd[format=glsignore]{\@glo@entry}}%
  }%
}
\newcommand*{\glsignore}[1]{%
  \edef\glsopenbrace{\expandafter\@gobble\string\{ }
  \edef\glsclosebrace{\expandafter\@gobble\string\} }
  \edef\glsbackslash{\expandafter\@gobble\string\ }
  \edef\glsquote#1{\string"#1\string"}
  \edef\glspercentchar{\expandafter\@gobble\string\% }
  \edef\glsstildechar{\string~ }
  \ifglsxindy
    \newcommand*{\@glsfirstletter}{A}
  \fi
  \newcommand*{\GlsSetXdyFirstLetterAfterDigits}{%
    \@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
  \ifglsxindy
    \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
      \renewcommand*{\@glsfirstletter}{#1}}
    \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
      \renewcommand*{\@glsfirstletter}{#1}%
      \@onelevel@sanitize\@glsfirstletter
    }
  \else
    \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
      \glsnoindywarning\GlsSetXdyFirstLetterAfterDigits}
    \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%
      \@GlsSetXdyFirstLetterAfterDigits
    }
  \fi
  \ifglsxindy
    \newcommand*{\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
  \fi
  \newcommand*{\GlsSetXdyNumberGroupOrder}[1]{%
    \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
  }
  \ifglsxindy
    \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
      \renewcommand*{\@xdynumbergrouporder}{#1}%
    }
    \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%

```

```

        \renewcommand*{\@xdynumbergrouporder}{#1}%
        \@onelevel@sanitize\@xdynumbergrouporder
    }
\else
    \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
        \glsnoxywarning\GlsSetXdyNumberGroupOrder}
    \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
        \@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*{\@glsmminrange}{2}
\ifglsxindy
    \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
        \renewcommand*{\@glsmminrange}{#1}}
\else
    \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
        \glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglsxindy
    \def\writeist{%
        \ifundef{\glswrite}{\newwrite\glswrite}{}%
        \@glscataddpredefinedattributes
        \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\@glsc@classI:=\@glsc@xdy@locationlist\do{%
            \protected@write\glswrite{ }(define-location-class
                \string"\@glsc@classI\string"^^J\space\space\space
                (
                    :sep "{ }{"
                    \csname @glsc@xdy@Lclass@\@glsc@classI\endcsname\space
                    :sep "}"
                )
            )
            ^^J\space\space\space

```

```

:min-range-length \@glsminrange^^J%
)
}%
{%
\@for\@gls@classII:=\@gls@xdy@locationlist\do{%
\protected@write\glswrite{}\{(define-location-class
\string"\@gls@classII-\@gls@classI\string"
^^J\space\space\space
(
:sep "{"
\csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
:sep "{{"
\csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
:sep "}"
)
^^J\space\space\space
:min-range-length \@glsminrange^^J%
)
}%
}%
}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
:unverified )}%
\write\glswrite{(markup-crossref-list
:class \string"see\string"^^J\space\space\space
:open \string"\string\glsseeformat\string"
:close \string"{}\string")}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
(\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
:open \string"\string
\glossarysection[\string\glossarytoctitle]{\string
\glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
{%
\@for\@this@attr:=\@xdyattributelist\do{%
\protected@write\glswrite{}\{\string\providecommand*%
\expandafter\string
\csname glsX\@this@ctr X\@this@attr\endcsname[2]%
{%
\string\setentrycounter
[\expandafter\@gobble\string\#1]{\@this@ctr}%
\expandafter\string

```



```

\csname\@this@attr\endcsname
{\expandafter\@gobble\string\#2}%
}%
}%
}%
}%
\write\glswrite{%
\string\begin
{theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
\space\space:close \string"\glpercentchar\glstildechar n\string
\end{theglossary}\string\glossarypostamble
\glstildechar n\string" ^^J\space\space\space
:tree)}}%
\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\glstildechar n\string"}}%
\write\glswrite{(markup-indexentry
:open \string"\string\relax \string\glsresetentrylist
\glstildechar n\string"}}%
\write\glswrite{(markup-locclass-list :open
\string"\glsopenbrace\string\glossaryentrynumbers
\glsopenbrace\string\relax\space \string"^^J\space\space\space
:sep \string", \string"
:close \string"\glsclosebrace\glsclosebrace\string"}}%
\write\glswrite{(markup-locref-list
:sep \string"\string\delimN\space\string"}}%
\write\glswrite{(markup-range
:sep \string"\string\delimR\space\string"}}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixF" :length 1 :ignore-end)}}%
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixFF" :length 2 :ignore-end)}}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\glstildechar n\string"}}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
:open-head \string"\string\glsgroupheading
\glsopenbrace\string"^^J\space\space\space
:close-head \string"\glsclosebrace\string"}}%

```

```

\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}%
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax
\ifundef{\glswrite}{\newwrite\glswrite}{}\relax
\openout\glswrite=\istfilename
\write\glswrite{\glspersentchar\space makeindex style file
created by the glossaries package}
\write\glswrite{\glspersentchar\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\glresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"glscpositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\fi
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{
\ifglxindy
\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@checkesobar####1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checklevel####1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@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\@gls@openfile\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*{\@gls@openfile}[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
}
\ifglxsindy
\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
\ifglxindy
\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@dodolistparser{\noexpand\DeclareListParser
{\noexpand\glsnumlistparser}{\delimN}}}%
\@gls@dodolistparser
\fi
\let\makenoidxglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
\let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
\ifglsautomake
\renewcommand*{\@gls@doautomake}{%
\@for\@gls@type:=\@glo@types\do{%
\ifdefempty{\@gls@type}{}%
{\@gls@automake{\@gls@type}}}%
}%
}%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}
\AtEndDocument{%
\warn@nomakeglossaries
\warn@noprintglossary
}
\newcommand*{\makenoidxglossaries}{%
\@domakeglossaries
{%
\renewcommand{\@gls@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
    \fi
  }%
}

```

```

        \fi
    \else
        \@glsopenfile{\glswrite}{\@glo@type}%
        \immediate\write\glswrite{%
            \expandafter\the
            \csname glo@\@glo@type @filetok\endcsname}%
        \immediate\closeout\glswrite
    \fi
}%
}
\newcommand*{\gls@glossary}[1]{%
    \@gls@glossary{#1}%
}
\newcommand*{\@gls@glossary}[2]{%
    \if@gls@debug
        \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
    \fi
}
\newcommand{\@gls@renewglossary}{%
    \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
    \let\@gls@renewglossary\@empty
}
\newcommand*{\gls@wrglossary}[2]{%
    \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]{%
    \ifglsexonlyfirst
        \ifglssused{#1}{#2}%
    \else
        #2%
    \fi
}

```

```

\fi
}
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
\newcommand*{\gls@disablepagerefexpansion}{%
\for\@gls@this:=\gls@protected@pagefmts\do
{%
\expandafter\let\@gls@this\relax
}%
}
\newcommand*{\gls@alphpage}{\@alph\c@page}
\newcommand*{\gls@Alphpage}{\@Alph\c@page}
\newcommand*{\gls@numberpage}{\number\c@page}
\newcommand*{\gls@arabicpage}{\@arabic\c@page}
\newcommand*{\gls@romanpage}{\romannumeral\c@page}
\newcommand*{\gls@Romanpage}{\@Roman\c@page}
\newcommand*{\glsaddprotectedpagefmt}[1]{%
\ea\pt\o\gls@protected@pagefmts{\,\expandonce{\csname gls#1page\endcsname}}%
\csedef{gls#1page}{\,\expandonce{\csname#1\endcsname}\noexpand\c@page}%
\ea\pt\o\@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]{%
\ifglslsclocations
\@do@esc@wrglossary{#1}%
\else
\@do@noesc@wrglossary{#1}%
\fi
}
\newcommand*{\@do@noesc@wrglossary}[1]{%
\expandafter\def\expandafter\@glslocref\expandafter{\theglentrycounter}%
\expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
\ifx\@glsHlocref\@glslocref
\def\@glo@counterprefix{%
\else
\protected@edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
{\@glslocref}{\@glsHlocref}%
}%
\@do@gls@getcounterprefix

```

```

\fi
\edef\@gls@label{\glsdetoklabel{#1}}%
\@do@wrglossary
}
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodstrue
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
\begingroup
\let\gls@orgthe\the
\let\gls@orgnumber\number
\let\gls@orgarabic@arabic
\let\gls@orgromannumeral\romannumeral
\let\gls@orgalph\@alph
\let\gls@orgAlph\@Alph
\let\gls@orgRoman\@Roman
\ifglswrallowprimitivemods
\def\gls@the##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
\def\the{\expandafter\gls@the}%
\def\gls@number##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
\def\number{\expandafter\gls@number}%
\fi
\def\@arabic##1{%
\ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\romannumeral##1{%
\ifx##1\c@page \gls@romanpage\else\gls@orgromannumeral##1\fi}%
\def\@Roman##1{%
\ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
\ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
\ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\@gls@locref{\theHglentrycounter}%
\endgroup
\@gls@checkmkidxchars\@gls@locref
\expandafter\ifx\theHglentrycounter\theHglentrycounter\relax
\def\@glo@counterprefix{%
\else
\protected@edef\@gls@Hlocref{\theHglentrycounter}%
\@gls@checkmkidxchars\@gls@Hlocref
\edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
{\@gls@locref}{\@gls@Hlocref}%
}%
\@do@gls@getcounterprefix
\fi
\edef\@gls@label{\glsdetoklabel{#1}}%
\@do@wrglossary

```



```

}
\newcommand*{\@@do@@wrglossary}{%
  \ifglxsindy
    \expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
    \def\@glo@range{}%
    \expandafter\if\@glo@prefix(\relax
      \def\@glo@range{:open-range}%
    \else
      \expandafter\if\@glo@prefix)\relax
      \def\@glo@range{:close-range}%
    \fi
  \fi
  \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
      :key (\csname glo@#1@index\endcsname)
      :xref (\string"\@gls@xref\string")
      :attr \string"see\string"
    )
  }%
\else
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    \string\glossaryentry{\csname glo@#1@index\endcsname
      \@gls@encapchar glsseeformat\@gls@xref}{Z}}%
\fi
}
\def\@gls@fixbraces#1#2#3\@nil{%
  \ifx#2[\relax
    \@gls@fixbraces#1#2#3\@end@fixbraces
  \else
    \def#1{{#2#3}}%
  \fi
}
\def\@gls@fixbraces#1[#2]#3\@end@fixbraces{%
  \def#1{[#2]{#3}}%
}
\newrobustcmd*{\glssee}[3][\seename]{%
  \@do@seeglossary{#2}{[#1]{#3}}
\newcommand*{\@glssee}[3][\seename]{%
  \glssee[#1]{#3}{#2}}
\newrobustcmd*{\glsseeformat}[3][\seename]{%
  \emph{#1} \glsseelist{#2}}
\newrobustcmd*{\glsseelist}[1]{%
  \let\@gls@dolast\relax
  \let\@gls@donext\relax
  \@for\@gls@thislabel:=#1\do{%
    \ifx\@xfor@nextelement\@nnil
      \@gls@dolast
    \else
      \@gls@donext
    \fi
    \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
    \let\@gls@dolast\glsseelastsep
    \let\@gls@donext\glsseesep
  }%
}
\newcommand*{\glsseelastsep}{\space\andname\space}
\newcommand*{\glsseesep}{, }

```

```

\newrobustcmd*{\glsseeitem}[1]{\glshyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*{\glsseeitemformat}[1]{\glstrytext{#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@noprntglossary}{}%
\ifcsundef{printglossary}{}%
{%
  \@gls@warnonglossdefined
  \undef\printglossary
}
\newcommand*{\@printgloss@checkexists@noignored}[2]{%
  \@ifglossaryexists{#1}%
  {#2}%
  {%
    \ifignoredglossary{#1}%
    {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
    {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
  }%
}
\newcommand*{\@printgloss@checkexists@allowignored}[2]{%
  \s@ifglossaryexists{#1}%
  {#2}%
  {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\newcommand*{\printglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@glossary}%
}
\newcommand*{\printglossaries}{%
  \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}
\newcommand*{\printnoidxglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@noidx@glossary}%
}
\newcommand*{\printnoidxglossaries}{%
  \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
\newcommand*{\@printgloss@setsort}{%
}
\newcommand*{\@gls@preglossaryhook}{%
}
\newcommand*{\@printglossary}[2]{%

```

```

\def\@glo@type{\glsdefaulttype}%
\def\glossarytitle{\csname @glo@type\@glo@type @title\endcsname}%
\def\glossarytoctitle{\glossarytitle}%
\let\org@glossarytitle\glossarytitle
\def\@glossarystyle{%
  \ifx\@glossary@default@style\relax
    \GlossariesWarning{No default glossary style provided \MessageBreak
      for the glossary '\@glo@type'. \MessageBreak
      Using deprecated fallback. \MessageBreak
      To fix this set the style with \MessageBreak
      \string\setglossarystyle\space or use the \MessageBreak
      style key=value option}%
  \fi
}%
\def\gls@dotoc@title{\glssettoctitle{\@glo@type}}%
\let\@org@glossaryentrynumbers\glossaryentrynumbers
\bgroup
\@printgloss@setsort
\setkeys{printgloss}{#1}%
\@printgloss@checkexists{\@glo@type}%
{%
  \ifx\glossarytitle\org@glossarytitle
  \else
    \expandafter\let\csname @glo@type\@glo@type @title\endcsname
      \glossarytitle
  \fi
  \let\currentglossary\@glo@type
  \let\org@glossaryentrynumbers\glossaryentrynumbers
  \let\glsnonextpages\@glsnonextpages
  \let\glsnextpages\@glsnextpages
  \let\nopostdesc\@nopostdesc
  \gls@dotoc@title
  \@glossarystyle
  \let\gls@org@glossaryentryfield\glossentry
  \let\gls@org@glossarysubentryfield\subglossentry
  \renewcommand{\glossentry}[1]{%
    \xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
    \gls@org@glossaryentryfield{##1}%
  }%
  \renewcommand{\subglossentry}[2]{%
    \xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
    \gls@org@glossarysubentryfield{##1}{##2}%
  }%
  \@gls@preglossaryhook
  #2%
}%
\egroup
\global\let\glossaryentrynumbers\@org@glossaryentrynumbers
\global\let\warn@noprintglossary\relax
}

```

```

\newcommand{\@print@glossary}{%
  \makeatletter
  \@input@{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
  \IfFileExists{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
  {}%
  {\null}%
  \ifglxindy
    \ifcsundef{@xdy@\@glo@type @language}%
    {%
      \edef\@do@auxoutstuff{%
        \noexpand\AtEndDocument{%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\providecommand\string\@xdylanguage[2]{}%
          }%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\@xdylanguage{\@glo@type}{\@xdy@main@language}%
          }%
        }%
      }%
    }%
    {%
      \edef\@do@auxoutstuff{%
        \noexpand\AtEndDocument{%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\providecommand\string\@xdylanguage[2]{}%
          }%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\@xdylanguage{\@glo@type}{\csname @xdy@\@glo@type
              @language\endcsname}%
          }%
        }%
      }%
    }%
  }%
  \@do@auxoutstuff
  \edef\@do@auxoutstuff{%
    \noexpand\AtEndDocument{%
      \noexpand\immediate\noexpand\write\@auxout{%
        \string\providecommand\string\@gls@codepage[2]{}%
      }%
      \noexpand\immediate\noexpand\write\@auxout{%
        \string\@gls@codepage{\@glo@type}{\@gls@codepage}%
      }%
    }%
  }%
  \@do@auxoutstuff
}
\fi
\renewcommand*{\@warn@nomakeglossaries}{%
  \GlossariesWarningNoLine{\string\makeglossaries\space
    hasn't been used,^^Jthe glossaries will not be updated}%
}%
}
\newcommand*{\@glo@sortentries}[2]{%
  \glosortentrieswarning
  \def\@glo@sortinglist{}%
  \def\@glo@sortinghandler{#1}%
  \edef\@glo@type{#2}%

```

```

\forlistcsloop{\@glo@do@sortentries}{@glsref@#2}%
\csdef{@glsref@#2}{}%
\@for\@this@label:=\@glo@sortinglist\do{%
  \xifinlistcs{\@this@label}{@glsref@#2}%
  {%
    \listcsxadd{@glsref@#2}{\@this@label}%
  }%
  \ifcsdef{@glo@sortingchildren@\@this@label}%
  {%
    \@glo@addchildren{#2}{\@this@label}%
  }%
}%
}
\newcommand*{\@glo@addchildren}[2]{%
  \bgroup
  \letcs{\@glo@childlist}{@glo@sortingchildren@#2}%
  \@for\@this@childlabel:=\@glo@childlist\do
  {%
    \xifinlistcs{\@this@childlabel}{@glsref@#1}%
    {%
      \listcsxadd{@glsref@#1}{\@this@childlabel}%
    }%
    \ifcsdef{@glo@sortingchildren@\@this@childlabel}%
    {%
      \@glo@addchildren{#1}{\@this@childlabel}%
    }%
    {%
      }%
    }%
  }%
  \egroup
}
\newcommand*{\@glo@do@sortentries}[1]{%
  \ifglshasparent{#1}%
  {%
    \edef\@glo@parent{\csuse{glo@\glsdetoklabel{#1}@parent}}%
    \ifcsundef{@glo@sortingchildren@\@glo@parent}%
    {%
      \csdef{@glo@sortingchildren@\@glo@parent}{}%
    }%
    {%
      }%
    }%
    \expandafter\@glo@sortedinsert
    \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
    \xifinlistcs{\@glo@parent}{@glsref@\@glo@type}%
    {%
      }%
    {%
      \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
    }%
  }%
}

```

```

    }%
  }%
  {%
    \@glo@sortedinsert{\@glo@sortinglist}{#1}%
  }%
}
\newcommand*{\@glo@sortedinsert}[2]{%
  \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
}%
\newcommand*{\@glo@sorthandler@word}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@letter}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@case}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@nocase}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sortmacro@word}[1]{%

```

```

\ifdefstring{\@glo@default@sorttype}{standard}%
{%
  \@glo@sortentries{\@glo@sorthandler@word}{#1}%
}%
{%
  \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=word]}{}%
}%
}
\newcommand*{\@glo@sortmacro@letter}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=letter]}{}%
  }%
}
\newcommand*{\@glo@sortmacro@standard}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \ifcsdef{\@glo@sorthandler@glorder}%
    {%
      \@glo@sortentries{\cuse{\@glo@sorthandler@glorder}}{#1}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\glorder’}{}%
    }%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=standard]}{}%
  }%
}
\newcommand*{\@glo@sortmacro@case}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@case}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=case]}{}%
  }%
}
\newcommand*{\@glo@sortmacro@nocase}[1]{%

```



```

\ifdefstring{\@glo@default@sorttype}{standard}%
{%
  \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
}%
{%
  \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=nocase]}{}%
}%
}
\newcommand*{\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{}%
  \forgl@sentries[#1]{\@gls@thislabel}%
  {%
    \xifinlistcs{\@gls@thislabel}{\@glsref@#1}%
    {%
      \listcsadd{\@glo@sortinglist}{\@gls@thislabel}%
    }%
    {%
      }%
    }%
  }%
  \cslet{\@glsref@#1}{\@glo@sortinglist}%
}
\newcommand*{\@glo@sortmacro@def@do}[1]{%
  \ifinlistcs{#1}{\@glsref@{\@glo@type}%
  }%
  {%
    \listcsadd{\@glsref@{\@glo@type}}{#1}%
  }%
  \ifcsdef{\@glo@sortingchildren@#1}%
  {%
    \@glo@addchildren{\@glo@type}{#1}%
  }%
  }%
}
\newcommand*{\@glo@sortmacro@use}[1]{%
  \newcommand*{\@print@noidx@glossary}{%
    \ifcsdef{\@glsref@{\@glo@type}%
    }%
    \ifcsdef{\@glo@sortmacro@{\@glo@sorttype}%
    }%
    \csuse{\@glo@sortmacro@{\@glo@sorttype}}{\@glo@type}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}%
    }%
    \glossarysection[\glossarytoctitle]{\glossarytitle}%
    \glossarypreamble
    \def\@gls@currentlettergroup{}%
    \begin{theglossary}%

```

```

\glossaryheader
\glsresetentrylist
\forlistcsloop{\@gls@noidx@do}{\@glsref@{\@glo@type}%
\end{theglossary}}%
\glossarypostamble
}%
{%
\@gls@noref@warn{\@glo@type}%
}%
}
\def\glo@grabfirst#1#2\@nil{%
\def\@gls@firsttok{#1}%
\ifdefempty\@gls@firsttok
{%
\def\@glo@thislettergrp{0}%
}%
{%
\@onelevel@sanitize\@gls@firsttok
\expandafter\@glo@grabfirst\@gls@firsttok{}}\@nil
}%
}
\def\@glo@grabfirst#1#2\@nil{%
\ifdefempty\@glo@thislettergrp
{%
\def\@glo@thislettergrp{glssymbols}%
}%
{%
\count@=\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@dotoc@title\relax
}
\define@key{printgloss}{toctitle}{%
\def\glossarytoctitle{#1}%
\let\gls@dotoc@title\relax
}
\define@key{printgloss}{style}{%
\ifcsundef{\@glsstyle@#1}%
{%
\PackageError{glossaries}%
{Glossary style ‘#1’ undefined}{}%
}%
{%
\def\@glossarystyle{\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*{\glssentryitem}[1]{%
  \ifglssentrycounter
    \glssstepentry{#1}\glssentrycounterlabel
  \else
    \glssresetsubentrycounter
  \fi
}
\newcommand*{\glsssubentryitem}[1]{%
  \ifglsssubentrycounter
    \glssstepsubentry{#1}\glsssubentrycounterlabel
  \fi
}
\ifcsundef{theglossary}%
{%
  \newenvironment{theglossary}{}{}%
}%
{%
  \@gls@warnontheGLOSSdefined
  \renewenvironment{theglossary}{}{}%
}
\newcommand*{\glossaryheader}{}
\newcommand*{\glstarget}[2]{\@glstarget{\glolinkprefix#1}{#2}}
\providecommand*{\compatibleglossentry}[2]{%
  \toks@{#2}%
  \protected@edef\@do@glossentry{\noexpand\glossaryentryfield{#1}%
    {\noexpand\glsnamefont
      {\expandafter\expandonce\csname glo@#1@name\endcsname}}}%
    {\expandafter\expandonce\csname glo@#1@desc\endcsname}%
    {\expandafter\expandonce\csname glo@#1@symbol\endcsname}%
    {\the\toks@}}%
  }%
  \@do@glossentry
}
\newcommand*{\glossentryname}[1]{%
  \glssdoifexistsorwarn{#1}%
  {%
    \letcs{\glo@name}{\glo@\glssdetoklabel{#1}@name}%
    \expandafter\glsnamefont\expandafter{\glo@name}%
  }%
}
\newcommand*{\Glossentryname}[1]{%
  \glssdoifexistsorwarn{#1}%
  {%
    \glsnamefont{\Glsentryname{#1}}%
  }%
}

```

```

\newcommand*{\glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrydesc{#1}%
  }%
}
\newcommand*{\Glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrydesc{#1}%
  }%
}
\newcommand*{\glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrysymbol{#1}%
  }%
}
\newcommand*{\Glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrysymbol{#1}%
  }%
}
\providecommand*{\compatiblesubglossentry}[3]{%
  \toks@{#3}%
  \protected@edef\@do@subglossentry{\noexpand\glossarysubentryfield{\number#1}%
    {#2}%
    {\noexpand\glsnamefont
      {\expandafter\expandonce\csname glo@#2@name\endcsname}}%
    {\expandafter\expandonce\csname glo@#2@desc\endcsname}%
    {\expandafter\expandonce\csname glo@#2@symbol\endcsname}%
    {\the\toks@}}%
  }%
  \@do@subglossentry
}
\newcommand*{\setglossentrycompatibility}{%
  \let\glossentry\compatibleglossentry
  \let\subglossentry\compatiblesubglossentry
}
\setglossentrycompatibility
\newcommand{\glossaryentryfield}[5]{%
  \GlossariesWarning
  {Deprecated use of \string\glossaryentryfield.^^J
    I recommend you change to \string\glossentry.^^J
    If you've just upgraded, try removing your gls auxiliary
    files^^J and recompile}%
  \noindent\textbf{\glstarget{#1}{#2}} #4 #3. #5\par}
\newcommand*{\glossarysubentryfield}[6]{%
  \GlossariesWarning

```



```

{Deprecated use of \string\glossarysubentryfield.^^J
 I recommend you change to \string\subglossentry.^^J
 If you've just upgraded, try removing your gls auxiliary
 files^^J and recompile}%
\glstarget{#2}{\strut}#4. #6\par}
\newcommand*{\glsgroupskip}{}
\newcommand*{\glsgroupheading}[1]{}
\newcommand*{\glsgetgrouptitle}[1]{%
 \@gls@getgrouptitle{#1}{\@gls@grptitle}%
 \@gls@grptitle
}
\newcommand*{\@gls@getgrouptitle}[2]{%
 \dtl@ifsingle{#1}%
 {%
 \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}}%
}%
{%
 \ifboolexpr{test{\ifstrequal{#1}{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][1]{%
 \def\@glo@counterprefix{#1}%
 \ifx\@glo@counterprefix\@empty
 \def\@glo@counterprefix{.}%
 \else
 \def\@glo@counterprefix{.#1}%
 \fi
 \def\glsentrycounter{#2}%
}
\newcommand*{\setglossarystyle}[1]{%
 \ifcsundef{@glsstyle@#1}%
 {%
 \PackageError{glossaries}{Glossary style '#1' undefined}{}%

```

```

}%
{%
  \csname @glsstyle@#1\endcsname
}%
\ifx\@glossary@default@style\relax
  \protected@edef\@glossary@default@style{#1}%
\fi
}
\newcommand*\@glossarystyle[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \GlossariesWarning
    {Deprecated command \string\glossarystyle.^^J
      I recommend you switch to \string\setglossarystyle\space unless
      you want to maintain backward compatibility}%
    \setglossentrycompatibility
    \csname @glsstyle@#1\endcsname
    \ifcsdef{@glscompstyle@#1}%
    {\setglossentrycompatibility\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\gslongtok
\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}%
  \gslongtok{##4}%
  \newacronymhook
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\expandonce{\acronymentry{##2}}},%
      sort={\acronymsort{\the\glsshorttok}{\the\gslongtok}},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\gslongtok},%
      longplural={\the\gslongtok\noexpand\acrpluralsuffix},%
      \GenericAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}%
\renewcommand*\@acrfullfmt}[3]{%
  \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}%
\renewcommand*\@Acrfullfmt}[3]{%

```

```

\glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}%
\renewcommand*{\acrfullplfmt}[3]{%
\glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\glsentryfull}[1]{\genacrfullformat{##1}{}}}%
\renewcommand*{\Glsentryfull}[1]{\Genacrfullformat{##1}{}}}%
\renewcommand*{\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}}%
\renewcommand*{\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}}%
}
\newcommand*{\GenericAcronymFields}{description={\the\glslongtok}}
\newcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{#1}}}
\newcommand*{\acronymsort}[2]{#1}
\newcommand*{\setacronymstyle}[1]{%
\ifcsundef{@glsacr@dispstyle@#1}
{%
\PackageError{glossaries}{Undefined acronym style ‘#1’}{}%
}%
{%
\ifdefempty{@glsacronymlists}%
{%
\DeclareAcronymList{\acronymtype}%
}%
{}%
\SetGenericNewAcronym
\GlsUseAcrStyleDefs{#1}%
\@for\@gls@type:=\@glsacronymlists\do{%
\defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{#1}}%
}%
}%
}
\newcommand*{\newacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@#1}%
{%
\PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
}%
{%
\csdef{@glsacr@dispstyle@#1}{#2}%
\csdef{@glsacr@styledefs@#1}{#3}%
}%
}
\newcommand*{\renewacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@#1}%
{%

```

```

\csdef{@glsacr@dispstyle@#1}{#2}%
\csdef{@glsacr@styledefs@#1}{#3}%
}%
{%
\PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
}%
}
\newcommand*{\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*{\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
{%
\ifglshaslong{\glslabel}{\glsacrfmt}{\glsacrentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
\ifglshaslong{\glslabel}{\glsacrfmt}{\glsacrentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}

```



```

}%
\renewcommand*{\genplacrfullformat}[2]{%
\glentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newcommand*{\glsacspace}[1]{%
\settowidth{\dimen@}{(\protect\firstacronymfont{\glentryshort{##1}})%}
\ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
\ifglshaslong{\glslabel}{\glsacspace}{\glsacspace}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\protect\firstacronymfont{\glentryshort{##1}}##2\space
(\glentrylong{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshort{##1}}##2\space
(\glentrylong{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\protect\firstacronymfont{\glentryshortpl{##1}}##2\space
(\glentrylongpl{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
(\glentrylongpl{##1})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{%
\GlsUseAcrEntryDisplayStyle{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}{\glssacrpluralsuffix}%
}
\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}{\glssacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sp-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sp-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sp-short}%
}

```

```

\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glentrylong{##1}\glspacespace{##1}(\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sc-short-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{long-sc-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sc-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sm-short-long-desc}%

```

```

{%
  \GlsUseAcrEntryDisplayStyle{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}{\glacrpluralsuffix}%
  \renewcommand*{\acrfullfmt}[3]{%
    \glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
      (\glsentrylong{##2})}%
  }%
  \renewcommand*{\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
      (\glsentrylong{##2})}%
  }%
  \renewcommand*{\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
      \mfirstucMakeUppercase{\acronymfont{\glsentryshort{##2}}##3\space
        (\glsentrylong{##2})}}%
  }%
  \renewcommand*{\acrfullplfmt}[3]{%
    \glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
      (\glsentrylongpl{##2})}%
  }%
  \renewcommand*{\Acrfullplfmt}[3]{%
    \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
      (\glsentrylongpl{##2})}%
  }%
  \renewcommand*{\ACRfullplfmt}[3]{%
    \glslink[##1]{##2}{%
      \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{##2}}##3\space
        (\glsentrylongpl{##2})}}%
  }%
  \renewcommand*{\glsentryfull}[1]{%
    \acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
  \renewcommand*{\Glsentryfull}[1]{%
    \acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%

```

```

\renewcommand*{\glentryfullpl}[1]{%
  \acronymfont{\glentryshortpl{##1}}\space(\glentrylongpl{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\glentrylongpl{##1})}%
}
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}%
\newacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}%
\newacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}

```

```

}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##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
}
\ifglssacrshortcuts
  \DefineAcronymSynonyms
\fi
\newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
  \defglentryfmt[#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:=\@gls@acronymlists\do{%
        \SetDefaultAcronymDisplayStyle{\@gls@type}%
    }%
    \renewcommand{\newacronym}[4][ ]{%
        \ifx\@gls@acronymlists\@empty
            \def\@glo@type{\acronymtype}%
            \setkeys{glossentry}{##1}%
            \DeclareAcronymList{\@glo@type}%
            \SetDefaultAcronymDisplayStyle{\@glo@type}%
        \fi
        \glskeylisttok{##1}%
        \glslabeltok{##2}%
        \glsshorttok{##3}%
        \glslongtok{##4}%
        \newacronymhook
        \DefaultNewAcronymDef
    }%
    \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
\newcommand*{\acrlinkfootnote}[3]{%
    \footnote{\glslink{#1}{#2}{#3}}%
}
\newcommand*{\acrno linkfootnote}[3]{%
    \footnote{#3}%
}
\newcommand*{\SetDescriptionFootnoteAcronymDisplayStyle}[1]{%
    \def\glsentryfmt[1]{%
        \ifdefempty\glscustomtext

```

```

{%
  \ifglused{\glslabel}%
  {%
    \acronymfont{\glsenentryfmt}%
  }%
  {%
    \firstacronymfont{\glsenentryfmt}%
    \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\@gls@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@gls@type}%
      \SetDescriptionFootnoteAcronymDisplayStyle{\@gls@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]{\glsentryfmt}%
}
\newcommand*{\DescriptionDUANewAcronymDef}{%
  \edef\do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glslongtok},%
      sort={\the\glslongtok},

```

```

text={\the\glslongtok},%
first={\the\glslongtok},%
plural={\noexpand\expandonce\noexpand\@glo@longpl},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
symbol={\the\glsshorttok},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionDUAAcronymStyle}{%
  \ifglsacrsmallcaps
    \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
      can’t both be set}{}%
  \else
    \ifglsacrsmaller
      \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
        can’t both be set}{}%
    \fi
  \fi
\renewcommand{\newacronym}[4][\]{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetDescriptionDUAAcronymDisplayStyle{\@glo@type}%
  \fi
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%

```

```

\glslongtok{##4}%
\newacronymhook
\DescriptionDUANewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetDescriptionDUAAcronymDisplayStyle{\@gls@type}%
}%
}%
\newcommand*{\SetDescriptionAcronymDisplayStyle}[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\ifglsused{\glslabel}%
{%
\let\gls@org@insert\glsinsert
\let\glsinsert\@empty
\acronymfont{\glsentryfmt}\gls@org@insert
}%
{%
\glsentryfmt
\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

```

```

\ifglsmaller
\renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
\fi
\fi
}%
\newcommand*{\SetFootnoteAcronymDisplayStyle}[1]{%
\defglentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\let\gls@org@insert\glsinsert
\let\glsinsert\@empty
\ifglused{\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
{\glentrylongpl{\glslabel}}%
{\glentrylong{\glslabel}}%
}%
}%
}%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*{\FootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
first={\the\glsshorttok},%
firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\the\glslongtok},%
descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%

```

```

        \the\glskeylisttok
    }%
}%
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
    \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
    \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
    \@@gls@expand@field{##1}{descplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@descplural\@org@gls@assign@descplural
}
\newcommand*{\SetFootnoteAcronymStyle}{%
    \renewcommand{\newacronym}[4][{}]{%
        \ifx\@glsacronymlists\@empty
            \def\@glo@type{\acronymtype}%
            \setkeys{glossentry}{##1}%
            \DeclareAcronymList{\@glo@type}%
            \SetFootnoteAcronymDisplayStyle{\@glo@type}%
        \fi
        \glskeylisttok{##1}%
        \glslabeltok{##2}%
        \glsshorttok{##3}%
        \glslongtok{##4}%
        \newacronymhook
        \FootnoteNewAcronymDef
    }%
    \@for\@gls@type:=\@glsacronymlists\do{%
        \SetFootnoteAcronymDisplayStyle{\@gls@type}%
    }%
    \ifglsacrsmallcaps
        \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
        \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
    \else
        \ifglsacrsmaller
            \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
        \fi
    \fi
    \ifglsacrdua
        \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
            can’t both be set}{}%
    \fi

```



```

}%
\DeclareRobustCommand*{\glsdoparenifnotempty}[2]{%
  \protected@edef\gls@tmp{#1}%
  \ifdefempty\gls@tmp
  {}%
  {%
    \ifx\gls@tmp\@gls@default@value
    \else
    \space (#2{#1})%
    \fi
  }%
}
}
\newcommand*{\SetSmallAcronymDisplayStyle}[1]{%
  \def\glsentryfmt[#1]{%
    \ifdefempty\glscustomtext
    {%
      \let\gls@org@insert\glsinsert
      \let\glsinsert\@empty
      \ifglsused{\glslabel}%
      {%
        \acronymfont{\glsentryfmt}\gls@org@insert
      }%
      {%
        \glsentryfmt
        \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@gl@assign@descplural\gl@assign@descplural
\def\gl@assign@firstpl##1##2{%
  \@@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@symbolplural##1##2{%
  \@@gl@expand@field{##1}{symbolplural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@@gl@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\org@gl@assign@firstpl
\let\gl@assign@plural\org@gl@assign@plural
\let\gl@assign@symbolplural\org@gl@assign@symbolplural
\let\gl@assign@descplural\org@gl@assign@descplural
}
\newcommand*{\SetDUASstyle}{%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@gl@acronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetDUADisplayStyle{\@glo@type}%
    \fi
    \gl@keylisttok{##1}%
    \gl@labeltok{##2}%
    \gl@shorttok{##3}%
    \gl@longtok{##4}%
    \newacronymhook
    \DUANewAcronymDef
  }%
  \@for\@gl@type:=\@gl@acronymlists\do{%
    \SetDUADisplayStyle{\@gl@type}%
  }%
}
\newcommand*{\SetAcronymStyle}{%
  \SetDefaultAcronymStyle
  \ifgl@acrdescription
    \ifgl@acrfootnote
      \SetDescriptionFootnoteAcronymStyle
    \else
      \ifgl@acrdua
        \SetDescriptionDUAAcronymStyle
      \else
        \SetDescriptionAcronymStyle
      \fi
    \fi
  \fi
}

```

```

\else
  \ifglacrfootnote
    \SetFootnoteAcronymStyle
  \else
    \ifthenelse{\boolean{glacrsmallcaps}\OR
      \boolean{glacrsmaller}}{%
      {%
        \SetSmallAcronymStyle
      }%
    }%
    \ifglacrdua
      \SetDUASstyle
    \fi
  }%
\fi
\fi
}
\SetAcronymStyle
\newcommand*{\SetCustomDisplayStyle}[1]{%
  \defglentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*{\CustomAcronymFields}{%
  name={\the\glsshorttok},%
  description={\the\glslongtok},%
  first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
  firstplural={\acrfullformat
    {\noexpand\glsentrylongpl{\the\glslabeltok}}%
    {\noexpand\glsentryshortpl{\the\glslabeltok}}},%
  text={\the\glsshorttok},%
  plural={\the\glsshorttok\noexpand\acrpluralsuffix}%
}
\newcommand*{\CustomNewAcronymDef}{%
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      user1={\the\glsshorttok},%
      user2={\the\glsshorttok\noexpand\acrpluralsuffix},%
      user3={\the\glslongtok},%
      user4={\the\glslongtok\noexpand\acrpluralsuffix},%
      \CustomAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}

```

```

\newcommand*{\SetCustomStyle}{%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@glsacronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetCustomDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \CustomNewAcronymDef
  }%
  \@for\@gls@type:=\@glsacronymlists\do{%
    \SetCustomDisplayStyle{\@gls@type}%
  }%
}
\RequirePackage{glossary-hypernav}[=v4.46]
\@gls@loadlist
\@gls@loadlong
\@gls@loadsuper
\@gls@loadtree
\ifx\@glossary@default@style\relax
\else
  \setglossarystyle{\@glossary@default@style}
\fi
\newcommand*{\showgloparent}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
}
\newcommand*{\showglolevel}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
}
\newcommand*{\showglotext}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
}
\newcommand*{\showgloplural}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@plural\endcsname
}
\newcommand*{\showglofirst}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@first\endcsname
}
\newcommand*{\showglofirstpl}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpl\endcsname
}
\newcommand*{\showglotype}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@type\endcsname
}
\newcommand*{\showglocounter}[1]{%

```

```

\expandafter\show\csname glo@\glsdetoklabel{#1}@counter\endcsname
}
\newcommand*{\showglouserii}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@userii\endcsname
}
\newcommand*{\showglouseriii}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@useriii\endcsname
}
\newcommand*{\showglouseriv}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@useriv\endcsname
}
\newcommand*{\showglouserv}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@userv\endcsname
}
\newcommand*{\showglouservi}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@uservi\endcsname
}
\newcommand*{\showgloname}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@name\endcsname
}
\newcommand*{\showglodesc}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@desc\endcsname
}
\newcommand*{\showglodescplural}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@descplural\endcsname
}
\newcommand*{\showglosort}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@sort\endcsname
}
\newcommand*{\showglosymbol}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@symbol\endcsname
}
\newcommand*{\showglosymbolplural}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@symbolplural\endcsname
}
\newcommand*{\showgloshort}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@short\endcsname
}
\newcommand*{\showglolong}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname
}
\newcommand*{\showgloindex}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname
}
\newcommand*{\showgloflag}[1]{%
\expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname
}

```

```

\newcommand*{\showgloclist}[1]{%
  \expandafter\show\csname glo@glstoklabel{#1}@loclist\endcsname
}
\newcommand*{\showglofield}[2]{%
  \csshow{glo@glstoklabel{#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 glist@#1\endcsname
}
\csname ifglsc-compatible-2.07\endcsname
  \RequirePackage{glossaries-compatible-207}[v4.46]
\fi

```

8.2 Rollback v4.46 (glossaries-accsupp-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-accsupp}[2020/03/19 v4.46 (NLCT)
  Experimental glossaries accessibility]
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
\ProcessOptions
\@ifpackageloaded{glossaries-extra}
{%
  \ifx@glsxtr@doaccsupp\empty
    \GlossariesWarning{The 'glossaries-accsupp'
      package has been loaded\MessageBreak
      after the 'glossaries-extra' package. This\MessageBreak
      can cause a failure to integrate both packages. \MessageBreak
      Either use the 'accsupp' option when you load\MessageBreak
      'glossaries-extra' or load 'glossaries-accsupp'\MessageBreak
      before loading 'glossaries-extra'}%
  }

```



```

\fi
}
{}
\def\compatibleglossentry#1#2{%
  \toks@{#2}%
  \protected@edef\@do@glossentry{%
    \noexpand\accsuppglossaryentryfield{#1}%
    {\noexpand\glsnamefont
      {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@name\endcsname}}}%
    {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@desc\endcsname}%
    {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@symbol\endcsname}%
    {\the\toks@}%
  }%
  \@do@glossentry
}
\def\compatiblesubglossentry#1#2#3{%
  \toks@{#3}%
  \protected@edef\@do@subglossentry{%
    \noexpand\accsuppglossarysubentryfield{\number#1}%
    {#2}%
    {\noexpand\glsnamefont
      {\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@name\endcsname}}}%
    {\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@desc\endcsname}%
    {\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@symbol\endcsname}%
    {\the\toks@}%
  }%
  \@do@subglossentry
}
\RequirePackage{glossaries}[v4.46]
\providecommand{\gls@accsupp@engine}{accsupp}
\providecommand{\gls@accessibility}[4]{#4}
\ifdefstring\gls@accsupp@engine{accsupp}
{
  \RequirePackage{accsupp}
  \renewcommand{\gls@accessibility}[4]{%
    \BeginAccSupp{#1,#2={#3}}#4\EndAccSupp{}}%
}
}
{}
\newcommand{\glsaccessibility}[4][[]]{%
  \@glsshowaccsupp{#1}{#2}{#3}%
  \gls@accessibility{#1}{#2}{#3}{#4}%
}
\define@key{glossentry}{access}{%
  \def\@glo@access{#1}%
}
\define@key{glossentry}{textaccess}{%
  \def\@glo@textaccess{#1}%
}
\define@key{glossentry}{firstaccess}{%

```

```

\def\@glo@firstaccess{#1}%
}
\define@key{glossentry}{pluralaccess}{%
\def\@glo@pluralaccess{#1}%
}
\define@key{glossentry}{firstpluralaccess}{%
\def\@glo@firstpluralaccess{#1}%
}
\define@key{glossentry}{symbolaccess}{%
\def\@glo@symbolaccess{#1}%
}
\define@key{glossentry}{symbolpluralaccess}{%
\def\@glo@symbolpluralaccess{#1}%
}
\define@key{glossentry}{descriptionaccess}{%
\def\@glo@descaccess{#1}%
}
\define@key{glossentry}{descriptionpluralaccess}{%
\def\@glo@descpluralaccess{#1}%
}
\define@key{glossentry}{shortaccess}{%
\def\@glo@shortaccess{#1}%
}
\define@key{glossentry}{shortpluralaccess}{%
\def\@glo@shortpluralaccess{#1}%
}
\define@key{glossentry}{longaccess}{%
\def\@glo@longaccess{#1}%
}
\define@key{glossentry}{longpluralaccess}{%
\def\@glo@longpluralaccess{#1}%
}
\define@key{glossentry}{user1access}{%
\def\@glo@useriaccess{#1}%
}
\define@key{glossentry}{user2access}{%
\def\@glo@useriiaccess{#1}%
}
\define@key{glossentry}{user3access}{%
\def\@glo@useriiiaccess{#1}%
}
\define@key{glossentry}{user4access}{%
\def\@glo@userivaccess{#1}%
}
\define@key{glossentry}{user5access}{%
\def\@glo@uservaccess{#1}%
}
\define@key{glossentry}{user6access}{%
\def\@glo@userviaccess{#1}%
}
}

```

```

\appto\@gls@keymap{,%
  {access}{access},%
  {textaccess}{textaccess},%
  {firstaccess}{firstaccess},%
  {pluralaccess}{pluralaccess},%
  {firstpluralaccess}{firstpluralaccess},%
  {symbolaccess}{symbolaccess},%
  {symbolpluralaccess}{symbolpluralaccess},%
  {descaccess}{descaccess},%
  {descpluralaccess}{descpluralaccess},%
  {shortaccess}{shortaccess},%
  {shortpluralaccess}{shortpluralaccess},%
  {longaccess}{longaccess},%
  {longpluralaccess}{longpluralaccess},%
  {user1access}{useriaccess},%
  {user2access}{useriiaccess},%
  {user3access}{useriiiaccess},%
  {user4access}{userivaccess},%
  {user5access}{uservaccess},%
  {user6access}{userviaccess}%
}
\def\@gls@noaccess{\relax}
\let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
\renewcommand*{\@newglossaryentryprehook}{%
  \@gls@oldnewglossaryentryprehook
  \def\@glo@access{\relax}%
  \def\@glo@textaccess{\@glo@access}%
  \def\@glo@firstaccess{\@glo@access}%
  \def\@glo@pluralaccess{\@glo@textaccess}%
  \def\@glo@firstpluralaccess{\@glo@pluralaccess}%
  \def\@glo@symbolaccess{\relax}%
  \def\@glo@symbolpluralaccess{\@glo@symbolaccess}%
  \def\@glo@descaccess{\relax}%
  \def\@glo@descpluralaccess{\@glo@descaccess}%
  \def\@glo@shortaccess{\relax}%
  \def\@glo@shortpluralaccess{\@glo@shortaccess}%
  \def\@glo@longaccess{\relax}%
  \def\@glo@longpluralaccess{\@glo@longaccess}%
  \def\@glo@useriaccess{\relax}%
  \def\@glo@useriiaccess{\relax}%
  \def\@glo@useriiiaccess{\relax}%
  \def\@glo@userivaccess{\relax}%
  \def\@glo@uservaccess{\relax}%
  \def\@glo@userviaccess{\relax}%
}
\let\@gls@oldnewglossaryentryposthook\@newglossaryentryposthook
\renewcommand*{\@newglossaryentryposthook}{%
  \@gls@oldnewglossaryentryposthook
  \expandafter
  \protected@xdef\csname glo@\@glo@label @access\endcsname{%

```

```

        \@glo@access}%
\expandafter
\protected@xdef\csname glo@\@glo@label @textaccess\endcsname{%
    \@glo@textaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstaccess\endcsname{%
    \@glo@firstaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @pluralaccess\endcsname{%
    \@glo@pluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstpluralaccess\endcsname{%
    \@glo@firstpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolaccess\endcsname{%
    \@glo@symbolaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolpluralaccess\endcsname{%
    \@glo@symbolpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descaccess\endcsname{%
    \@glo@descaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descpluralaccess\endcsname{%
    \@glo@descpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortaccess\endcsname{%
    \@glo@shortaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortpluralaccess\endcsname{%
    \@glo@shortpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longaccess\endcsname{%
    \@glo@longaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longpluralaccess\endcsname{%
    \@glo@longpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriaccess\endcsname{%
    \@glo@useriaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiaccess\endcsname{%
    \@glo@useriiaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiiaccess\endcsname{%
    \@glo@useriiiaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @userivaccess\endcsname{%
    \@glo@userivaccess}%
\expandafter

```

```

\protected@xdef\csname glo@\glo@label @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

```

```

        \xglsacccsupp{\@glo@access}{#1}%
    \fi
}
\newcommand*{\@gls@fieldaccess@display}[4]{%
    \protected@edef\@glo@access{#4}%
    \ifdefequal\@glo@access\@gls@noaccess
    {#3}%
    {\expandafter\glsfieldacccsupp\expandafter{\@glo@access}{#3}{#2}{#1}}%
}
\newrobustcmd*{\@gls@nameaccessdisplay}[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@access}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{name}{#1}{\@glsentryaccess{#2}}}%
}
\newrobustcmd*{\@gls@textaccessdisplay}[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@textaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{text}{#1}{\@glsentrytextaccess{#2}}}%
}
\newrobustcmd*{\@gls@pluralaccessdisplay}[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@pluralaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{plural}{#1}{\@glsentrypluralaccess{#2}}}%
}
\newrobustcmd*{\@gls@firstaccessdisplay}[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@firstaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{first}{#1}{\@glsentryfirstaccess{#2}}}%
}
\newrobustcmd*{\@gls@firstpluralaccessdisplay}[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@firstpluralaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{firstpl}{#1}{\@glsentryfirstpluralaccess{#2}}}%
}
\newrobustcmd*{\@gls@symbolaccessdisplay}[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@symbolaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{symbol}{#1}{\@glsentrysymbolaccess{#2}}}%
}
\newrobustcmd*{\@gls@symbolpluralaccessdisplay}[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@symbolpluralaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{symbolplural}{#1}{\@glsentrysymbolpluralaccess{#2}}}%
}
\newrobustcmd*{\@gls@descriptionaccessdisplay}[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@descaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{desc}{#1}{\@glsentrydescaccess{#2}}}%
}
\newrobustcmd*{\@gls@descriptionpluralaccessdisplay}[2]{%

```

```

\ifcsundef{glo@glsdetoklabel{#2}@descpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{descplural}{#1}{\glstrydescpluralaccess{#2}}}%
}
\newrobustcmd*{\glsshortaccessdisplay}[2]{%
\ifcsundef{glo@glsdetoklabel{#2}@shortaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{short}{#1}{\glstryshortaccess{#2}}}%
}
\newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
\ifcsundef{glo@glsdetoklabel{#2}@shortpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glstryshortpluralaccess{#2}}}%
}
\newrobustcmd*{\glslongaccessdisplay}[2]{%
\ifcsundef{glo@glsdetoklabel{#2}@longaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{long}{#1}{\glstrylongaccess{#2}}}%
}
\newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
\ifcsundef{glo@glsdetoklabel{#2}@longpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{longpl}{#1}{\glstrylongpluralaccess{#2}}}%
}
\newrobustcmd*{\glssuseriaccessdisplay}[2]{%
\ifcsundef{glo@glsdetoklabel{#2}@useriaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useri}{#1}{\glstryuseriaccess{#2}}}%
}
\newrobustcmd*{\glssuseriiaccessdisplay}[2]{%
\ifcsundef{glo@glsdetoklabel{#2}@useriiaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{userii}{#1}{\glstryuseriiaccess{#2}}}%
}
\newrobustcmd*{\glssuseriiaaaccessdisplay}[2]{%
\ifcsundef{glo@glsdetoklabel{#2}@useriiaaaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useriiaa}{#1}{\glstryuseriiaaaccess{#2}}}%
}
\newrobustcmd*{\glssuserivaccessdisplay}[2]{%
\ifcsundef{glo@glsdetoklabel{#2}@userivaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useriv}{#1}{\glstryuserivaccess{#2}}}%
}
\newrobustcmd*{\glssuservaccessdisplay}[2]{%
\ifcsundef{glo@glsdetoklabel{#2}@uservaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{userv}{#1}{\glstryuservaccess{#2}}}%
}
\newrobustcmd*{\glssuserviaccessdisplay}[2]{%

```



```

\ifcsundef{glo@glstetoklabel{#2}@userviaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{uservi}{#1}{\glstentryuserviaccess{#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
{\glstentryplural{\glslabel}}{\glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glstentrydescplural{\glslabel}}{\glslabel}}%
{\glssymbolpluralaccessdisplay
{\glstentrysymbolplural{\glslabel}}{\glslabel}}%
{\glsinsert}}%
}%
}%
#1{\glsfirstpluralaccessdisplay
{\glstentryfirstplural{\glslabel}}{\glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glstentrydescplural{\glslabel}}{\glslabel}}%
{\glssymbolpluralaccessdisplay
{\glstentrysymbolplural{\glslabel}}{\glslabel}}%
{\glsinsert}}%
}%
}%
}%
\ifglsused\glslabel
{%
#2{\glspluralaccessdisplay
{\Glsentryplural{\glslabel}}{\glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glstentrydescplural{\glslabel}}{\glslabel}}%
{\glssymbolpluralaccessdisplay
{\glstentrysymbolplural{\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}}%
      }%
    }%
  }%
}

```

```

        {\glsinsert}}}%
    }%
}%
}%
}%
{\%
\ifglsused{\glslabel}%
{\%
    #2{\glscustomtext}%
    {\glsdescriptionaccessdisplay
     {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
     {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
}%
{\%
    #1{\glscustomtext}%
    {\glsdescriptionaccessdisplay
     {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
     {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
}%
}%
}
\renewcommand*{\glsгенentryfmt}{\%
\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{\glentrytext{\glslabel}}}{\glslabel}%
\mfirstucMakeUppercase{\glsinsert}%
}%
{%
\glsfirstaccessdisplay
  {\mfirstucMakeUppercase{\glentryfirst{\glslabel}}}{\glslabel}%
\mfirstucMakeUppercase{\glsinsert}%
}%
}%
}%
}%
{\%
\glscustomtext\glsinsert
}%
}
\renewcommand*{\glsgenacfmt}{%
\ifdefempty\glscustomtext
{%
\ifglused\glslabel
{%
\glsifplural
{%
\glscapscase
{%
\acronymfont
  {\glsshortpluralaccessdisplay
    {\glentryshortpl{\glslabel}}{\glslabel}}%
\glsinsert
}%
{%
\acronymfont
  {\glsshortpluralaccessdisplay
    {\Glsentryshortpl{\glslabel}}{\glslabel}}%
\glsinsert
}%
{%
\mfirstucMakeUppercase
  {\acronymfont
    {\glsshortpluralaccessdisplay
      {\glentryshortpl{\glslabel}}{\glslabel}}%
\glsinsert}%
}%
}%
}%
\glscapscase
{%
\acronymfont
  {\glsshortaccessdisplay{\glentryshort{\glslabel}}{\glslabel}}%
\glsinsert

```

```

}%
{%
  \acronymfont
    {\glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
  \glsinsert
}%
{%
  \mfirstucMakeUppercase
    {\acronymfont{%
      \glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
    \glsinsert}%
}%
}%
}%
{%
  \glsifplural
  {%
    \glscapscase
    {%
      \genplacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
      \Genplacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
      \mfirstucMakeUppercase
        {\genplacrfullformat{\glslabel}{\glsinsert}}%
    }%
  }%
}%
{%
  \glscapscase
  {%
    \genacrfullformat{\glslabel}{\glsinsert}%
  }%
  {%
    \Genacrfullformat{\glslabel}{\glsinsert}%
  }%
  {%
    \mfirstucMakeUppercase
      {\genacrfullformat{\glslabel}{\glsinsert}}%
  }%
}%
}%
}%
}%
\glscustomtext
}%
}
\renewcommand*{\genacrfullformat}[2]{%
  \glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space

```

```

(\glsshortaccessdisplay{\protect\firstacronymfont{\glentryshort{#1}}}{#1})%
}
\renewcommand*{\Genacrfullformat}[2]{%
  \glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space
  (\glsshortaccessdisplay{\protect\firstacronymfont{\glentryshort{#1}}}{#1})%
}
\renewcommand*{\genplacrfullformat}[2]{%
  \glslongpluralaccessdisplay{\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@gl@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\glentryshort{#2}}{#2}}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@Acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@ACRshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree

```



```

\let\glsinsert\@empty
\def\glscustomtext{%
  \acronymfont{\glsshortaccessdisplay
    {\MakeUppercase{\glsentryshort{#2}}}{#2}}#3%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{\glsentrylong{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@AcrLong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{\Glsentrylong{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@ACRLong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{%
        \MakeUppercase{\glsentrylong{#2}}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%

```

```

\glspostlinkhook
}
\def\@glstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\glstrytext{#2}}{#2}#3}%
}
\def\@GLstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\Glstrytext{#2}}{#2}#3}%
}
\def\@GLStext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glstextaccessdisplay{\mfirstucMakeUppercase{\glstrytext{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\glstryfirst{#2}}{#2}#3}%
}
\def\@GLsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\Glstryfirst{#2}}{#2}#3}%
}
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsfirstaccessdisplay{\mfirstucMakeUppercase{\glstryfirst{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\glstryplural{#2}}{#2}#3}%
}
\def\@GLsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\Glstryplural{#2}}{#2}#3}%
}
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glspluralaccessdisplay{\mfirstucMakeUppercase{\glstryplural{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\glstryfirstplural{#2}}{#2}#3}%
}
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\Glstryfirstplural{#2}}{#2}#3}%
}
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsfirstpluralaccessdisplay{\mfirstucMakeUppercase{\glstryfirstplural{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glstryname{#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\@Glsymbol@#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\@Glsymbolplural@#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\@GLSuser@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuser@accessdisplay{\mfirstucMakeUppercase{\glsentryuser{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseri@accessdisplay{\glsentryuseri{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseri@accessdisplay{\Glsentryuseri{#2}}{#2}#3}%
}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuseri@accessdisplay{\mfirstucMakeUppercase{\glsentryuseri{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\glsentryname{#1}}{#1}}%
  }%
}
\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\Glsentryname{#1}}{#1}}%
  }%
}
\renewcommand*{\glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\glsentrydesc{#1}}{#1}%
  }%
}
\renewcommand*{\Glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}
\renewcommand*{\glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\glsentrysymbol{#1}}{#1}%
  }%
}
\renewcommand*{\Glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%

```

```

        \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
    }%
}
\newcommand*{\accsuppglossaryentryfield}[5]{%
    \glossaryentryfield{#1}%
    {\glslnameaccessdisplay{#2}{#1}}%
    {\glsldescriptionaccessdisplay{#3}{#1}}%
    {\glssymbolaccessdisplay{#4}{#1}}{#5}%
}
\newcommand*{\accsuppglossarysubentryfield}[6]{%
    \glossarysubentryfield{#1}{#2}%
    {\glslnameaccessdisplay{#3}{#2}}%
    {\glsldescriptionaccessdisplay{#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}%
{%
\GlsUseAcrEntryDisplayStyle{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}%
{%
\GlsUseAcrEntryDisplayStyle{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}}{##2}})%
\renewcommand*{\ACRfullplfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortpluralaccessdisplay
            {\mfirstucMakeUppercase
                {\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
                (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2}})}}%
\renewcommand*{\glsentryfull}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}\space
    (\glslongaccessdisplay{\glsentrylong{##1}}{##1}})%
\renewcommand*{\Glsentryfull}[1]{%
    \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##1}}}{##1}\space
    (\glslongaccessdisplay{\glsentrylong{##1}}{##1}})%
\renewcommand*{\glsentryfullpl}[1]{%
    \glsshortpluralaccessdisplay
        {\acronymfont{\glsentryshortpl{##1}}}{##1}\space
        (\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}})%
\renewcommand*{\Glsentryfullpl}[1]{%
    \glsshortpluralaccessdisplay
        {\acronymfont{\Glsentryshortpl{##1}}}{##1}\space
        (\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}})%
}
\renewacronymstyle{footnote-sc}%
{%
    \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
    \GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}

```

```

\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glstextup{\glspluralsuffix}}%
}%
\renewacronymstyle{footnote-sm}%
{%
\GlsUseAcrEntryDispStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}%
\renewacronymstyle{footnote-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sc-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
\GlsUseAcrStyleDefs{footnote-sc}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sm-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote-sm}%
}%
{%
\GlsUseAcrStyleDefs{footnote-sm}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
}

```

```

\newcommand*{\glsdefaultshortaccess}[2]{#1}
\renewcommand*{\newacronymhook}{%
  \edef\@gls@keylist{%
    shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    shortpluralaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    \the\glskeylisttok}%
  \expandafter\glskeylisttok\expandafter{\@gls@keylist}%
}
\renewcommand*{\DefaultNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      description={\the\glslongtok},%
      descriptionaccess=\relax,%
      text={\the\glsshorttok},%
      access={\noexpand\@glo@textaccess},%
      sort={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      first={\noexpand\glslongaccessdisplay
        {\the\glslongtok}{\the\glslabeltok}\space
        (\noexpand\glsshortaccessdisplay
          {\the\glsshorttok}{\the\glslabeltok})},%
      plural={\the\glsshorttok\acrpluralsuffix},%
      firstplural={\noexpand\glslongpluralaccessdisplay
        {\noexpand\@glo@longpl}{\the\glslabeltok}\space
        (\noexpand\glsshortpluralaccessdisplay
          {\noexpand\@glo@shortpl}{\the\glslabeltok})},%
      firstaccess=\relax,%
      firstpluralaccess=\relax,%
      textaccess={\noexpand\@glo@shortaccess},%
      \the\glskeylisttok
    }%
  }%
  \let\@org@gls@assign@firstpl\gls@assign@firstpl
  \let\@org@gls@assign@plural\gls@assign@plural
  \let\@org@gls@assign@descplural\gls@assign@descplural
  \def\gls@assign@firstpl##1##2{%
    \@gls@expand@field{##1}{firstpl}{##2}%
  }%
  \def\gls@assign@plural##1##2{%
    \@gls@expand@field{##1}{plural}{##2}%
  }%
  \def\gls@assign@descplural##1##2{%

```

```

    @@gls@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\DescriptionFootnoteNewAcronymDef}{%
\edef\do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
text={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
access={\noexpand\@glo@textaccess},%
plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
symbol={\the\glslongtok},%
symbolplural={\the\glslongtok\noexpand\acrpluralsuffix},%
firstpluralaccess=\relax,
textaccess={\noexpand\@glo@shortaccess},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\DescriptionNewAcronymDef}{%
\edef\do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%

```

```

type=\acronymtype,%
name={\noexpand
  \acronymformat{\the\glsshorttok}{\the\glslongtok}},%
access={\noexpand\@glo@textaccess},%
sort={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
first={\the\glslongtok},%
firstaccess=\relax,
firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
text={\the\glsshorttok},%
textaccess={\the\glslongtok},%
plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
symbol={\noexpand\@glo@text},%
symbolaccess={\noexpand\@glo@textaccess},%
symbolplural={\noexpand\@glo@plural},%
firstpluralaccess=\relax,
textaccess={\noexpand\@glo@shortaccess},%
\the\glskeylisttok}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\FootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\glsshorttok}},%
    sort={\the\glsshorttok},%
    text={\the\glsshorttok},%
    textaccess={\the\glslongtok},%
    access={\noexpand\@glo@textaccess},%

```



```

plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\the\glslongtok},%
descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
\the\glskeylisttok
}%
}%
\let\org@gls@assign@plural\gls@assign@plural
\let\org@gls@assign@firstpl\gls@assign@firstpl
\let\org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@gls@expand@field{##1}{descplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@plural\org@gls@assign@plural
\let\gls@assign@firstpl\org@gls@assign@firstpl
\let\gls@assign@descplural\org@gls@assign@descplural
}
\renewcommand*\SmallNewAcronymDef{%
\edef\@do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\glsshorttok}},%
    access={\noexpand\@glo@symbolaccess},%
    sort={\the\glsshorttok},%
    short={\the\glsshorttok},%
    shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
    shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    long={\the\glslongtok},%
    longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    text={\noexpand\@glo@short},%
    textaccess={\noexpand\@glo@shortaccess},%
    plural={\noexpand\@glo@shortpl},%
    first={\the\glslongtok},%
    firstaccess=\relax,
    firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    description={\noexpand\@glo@first},%
    descriptionplural={\noexpand\@glo@firstplural},%
    symbol={\the\glsshorttok},%
    symbolaccess={\the\glslongtok},%

```

```

symbolplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
\the\glskeylisttok
}%
}%
\let\org@gl@assign@firstpl\gl@assign@firstpl
\let\org@gl@assign@plural\gl@assign@plural
\let\org@gl@assign@descplural\gl@assign@descplural
\let\org@gl@assign@symbolplural\gl@assign@symbolplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@gl@expand@field{##1}{descplural}{##2}%
}%
\def\gl@assign@symbolplural##1##2{%
  \@gl@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\org@gl@assign@firstpl
\let\gl@assign@plural\org@gl@assign@plural
\let\gl@assign@descplural\org@gl@assign@descplural
\let\gl@assign@symbolplural\org@gl@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*{\showglofirstaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstaccess\endcsname
}
\newcommand*{\showglofirstpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpluralaccess\endcsname
}
\newcommand*{\showglofirstpluralsymbolaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpluralsymbolaccess\endcsname
}
\newcommand*{\showglofirstsymbolaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstsymbolaccess\endcsname
}
\newcommand*{\showglofirstsymbolpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstsymbolpluralaccess\endcsname
}

```

```

}
\newcommand*{\showgloDESCaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@DESCaccess\endcsname
}
\newcommand*{\showgloDESCpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@DESCpluralaccess\endcsname
}
\newcommand*{\showgloSHORTaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@SHORTaccess\endcsname
}
\newcommand*{\showgloSHORTpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@SHORTpluralaccess\endcsname
}
\newcommand*{\showgloLONGaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@LONGaccess\endcsname
}
\newcommand*{\showgloLONGpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@LONGpluralaccess\endcsname
}

```

8.3 Rollback v4.46 (glossaries-babel-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-babel}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
  }%
}%

```

8.4 Rollback v4.46 (glossaries-compatible-207-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-207}[2020/03/19 v4.46 (NLCT)]
\ifglxsindy
  \renewcommand*\GlsAddXdyAttribute[1]{%
    \edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string"}%
    \expandafter\toks@\expandafter{\@xdylocref}%
    \edef\@xdylocref{\the\toks@ ^^J}%
    (markup-locref
    :open \string"\string~\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{glshypernumber}
\GlsAddXdyAttribute{hyperrm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hypertt}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}
\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}
\ifglxsindy
  \renewcommand*\{GlsAddXdyLocation}[2]{%
    \edef\@xdyuserlocationdefs{%
      \@xdyuserlocationdefs ^^J%
      (define-location-class \string"#1\string"^^J\space\space
        \space(#2))
  }

```

```

}%
\edef\@xdyuserlocationnames{%
  \@xdyuserlocationnames^^J\space\space\space
  \string"#1\string"}%
}
\fi
\renewcommand{\@do@wrglossary}[1]{%
\ifglxsindy
  \expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
  \def\@glo@range{}%
  \expandafter\if\@glo@prefix(\relax
  \def\@glo@range{:open-range}%
  \else
  \expandafter\if\@glo@prefix\relax
  \def\@glo@range{:close-range}%
  \fi
\fi
\protected@edef\@glslocref{\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%
}
\ifglxsindy
  \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"\glscpositor\string"
    \string"arabic-numbers\string")
    :min-range-length \@glsmrange)}}%
\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\glssseeformat\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~n\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\glresetentrylist
    \string~n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsoopenbrace\string\glossaryentrynumbers
    \glsoopenbrace\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"glssymbols\string"}
\write\glswrite{page_compositor \string"glscpositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\noist
}
\fi
\renewcommand*{\noist}{\let\writeist\relax}

```

8.5 Rollback v4.46 (glossaries-compatible-307-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-307}[2020/03/19 v4.46 (NLCT)]
\newcommand{\compatglossarystyle}[2]{%
\ifcsundef{@glscompstyle@#1}%
{%
\csdef{@glscompstyle@#1}{#2}%
}%
}

```

```

    {%
      \PackageError{glossaries}{Glossary compatibility style ‘#1’ is already defined}{}%
    }%
  }
\compatglossarystyle{inline}{%
  \renewcommand{\glossaryentryfield}[5]{%
    \glsinlinedopostchild
    \gls@inlinesep
    \def\glo@desc{##3}%
    \def\@no@post@desc{\nopostdesc}%
    \glsentryitem{##1}\glsinlinenameformat{##1}{##2}%
    \ifx\glo@desc\@no@post@desc
      \glsinlineemptydescformat{##4}{##5}%
    \else
      \ifstreempty{##3}%
        {\glsinlineemptydescformat{##4}{##5}}%
        {\glsinlinedescformat{##3}{##4}{##5}}%
      \fi
      \ifglshaschildren{##1}%
        {%
          \glsresetsubentrycounter
          \glsinlineparentchildseparator
          \def\gls@inlinesubsep{}%
          \def\gls@inlinepostchild{\glsinlinepostchild}%
        }%
      {}%
    \def\gls@inlinesep{\glsinlineseparator}%
  }%
\renewcommand{\glossarysubentryfield}[6]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{##3}%
  \glssubentryitem{##2}\glsinlinesubdescformat{##4}{##5}{##6}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%
}
\compatglossarystyle{list}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item[\glsentryitem{##1}\glstarget{##1}{##2}]
      ##3\glspostdescription\space ##5}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6.}%
  }
\compatglossarystyle{listgroup}{%
  \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{listhypergroup}{%
  \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{altlist}{%

```

```

\renewcommand*{\glossaryentryfield}[5]{%
  \item[\glentryitem{##1}\glstarget{##1}{##2}]%
  \mbox{}\par\nobreak\@afterheading
  ##3\glspostdescription\space ##5}%
\renewcommand{\glossarysubentryfield}[6]{%
  \par
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}##4\glspostdescription\space ##6}%
}%
\compatglossarystyle{altlistgroup}{%
  \csuse{@glscmpstyle@altlist}%
}%
\compatglossarystyle{altlisthypergroup}{%
  \csuse{@glscmpstyle@altlist}%
}%
\compatglossarystyle{listdotted}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item[]\makebox[\glslistdottedwidth][l]{%
      \glentryitem{##1}\glstarget{##1}{##2}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##3}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    \item[]\makebox[\glslistdottedwidth][l]{%
      \glssubentryitem{##2}%
      \glstarget{##2}{##3}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##4}%
  }%
\compatglossarystyle{sublistdotted}{%
  \csuse{@glscmpstyle@listdotted}%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item[\glentryitem{##1}\glstarget{##1}{##2}]}%
}%
\compatglossarystyle{long}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
  }%
\compatglossarystyle{longborder}{%
  \csuse{@glscmpstyle@long}%
}%
\compatglossarystyle{longheader}{%
  \csuse{@glscmpstyle@long}%
}%
\compatglossarystyle{longheaderborder}{%
  \csuse{@glscmpstyle@long}%
}%
\compatglossarystyle{long3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%

```

```

\glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4 & ##6\\}%
}%
\compatglossarystyle{long3colborder}{%
\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long3colheader}{%
\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long3colheaderborder}{%
\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long4col}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
}%
\compatglossarystyle{long4colheader}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{long4colborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{long4colheaderborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4col}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colheader}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colheaderborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{longragged}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&

```

```

        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}##4\glspostdescription\space ##6%
        \tabularnewline}%
}%
\compatglossarystyle{longraggedborder}{%
  \csuse{@glscmpstyle@longragged}%
}%
\compatglossarystyle{longraggedheader}{%
  \csuse{@glscmpstyle@longragged}%
}%
\compatglossarystyle{longraggedheaderborder}{%
  \csuse{@glscmpstyle@longragged}%
}%
\compatglossarystyle{longragged3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glssubentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{longragged3colborder}{%
  \csuse{@glscmpstyle@longragged3col}%
}%
\compatglossarystyle{longragged3colheader}{%
  \csuse{@glscmpstyle@longragged3col}%
}%
\compatglossarystyle{longragged3colheaderborder}{%
  \csuse{@glscmpstyle@longragged3col}%
}%
\compatglossarystyle{altlongragged4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glssubentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%
\compatglossarystyle{altlongragged4colheader}{%
  \csuse{@glscmpstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colborder}{%
  \csuse{@glscmpstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colheaderborder}{%
  \csuse{@glscmpstyle@altlong4col}%
}%
\compatglossarystyle{index}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item\glssubentryitem{##1}\textbf{\glstarget{##1}{##2}}%

```

```

\ifx\relax##4\relax
\else
\space{##4}%
\fi
\space ##3\glspostdescription \space ##5}%
\renewcommand*{\glossarysubentryfield}[6]{%
\ifcase##1\relax
% level 0
\item
\or
% level 1
\subitem
\glssubentryitem{##2}%
\else
% all other levels
\subsubitem
\fi
\textbf{\glstarget{##2}{##3}}%
\ifx\relax##5\relax
\else
\space{##5}%
\fi
\space##4\glspostdescription\space ##6}%
}%
\compatglossarystyle{indexgroup}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{indexhypergroup}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{tree}{%
\renewcommand{\glossaryentryfield}[5]{%
\hangindent0pt\relax
\parindent0pt\relax
\glssentryitem{##1}\textbf{\glstarget{##1}{##2}}%
\ifx\relax##4\relax
\else
\space{##4}%
\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\textbf{\glstarget{##2}{##3}}%
\ifx\relax##5\relax
\else
\space{##5}%

```

```

\fi
\space##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treegroup}{%
\csuse{@glscmpstyle@tree}%
}%
\compatglossarystyle{treehypergroup}{%
\csuse{@glscmpstyle@tree}%
}%
\compatglossarystyle{treenoname}{%
\renewcommand{\glossaryentryfield}[5]{%
\hangindent0pt\relax
\parindent0pt\relax
\glstentryitem{##1}\textbf{\glstarget{##1}{##2}}%
\ifx\relax##4\relax
\else
\space(##4)%
\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\glstarget{##2}{\strut}%
##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treenonamegroup}{%
\csuse{@glscmpstyle@treenoname}%
}%
\compatglossarystyle{treenonamehypergroup}{%
\csuse{@glscmpstyle@treenoname}%
}%
\compatglossarystyle{alttree}{%
\renewcommand{\glossaryentryfield}[5]{%
\ifnum\@gls@prevlevel=0\relax
\else
\settowidth{\glstreeindent}{\textbf{\@glswidestname\space}}%
\hangindent\glstreeindent
\parindent\glstreeindent
\fi
\makebox[Opt][r]{\makebox[\glstreeindent][l]{%
\glstentryitem{##1}\textbf{\glstarget{##1}{##2}}}%
\ifx\relax##4\relax
\else
(##4)\space
\fi
##3\glspostdescription \space ##5\par
\def\@gls@prevlevel{0}%

```

```

}%
\renewcommand{\glossarysubentryfield}[6]{%
  \ifnum##1=1\relax
    \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{almtreegroup}{%
  \csuse{\@glscompstyle@almtree}%
}%
\compatglossarystyle{almtreehypergroup}{%
  \csuse{\@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolindex}{%
  \csuse{\@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexgroup}{%
  \csuse{\@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexhypergroup}{%

```



```

\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcoltree}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenoname}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenonamegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenonamehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolalmtree}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolalmtreegroup}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolalmtreehypergroup}{%
\csuse{@glscompstyle@almtree}%
}%

\compatglossarystyle{superragged}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glstentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4\glspostdescription\space ##6%
\tabularnewline}%
}%
\compatglossarystyle{superraggedborder}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheader}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheaderborder}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superragged3col}{%

```

```

\renewcommand*{\glossaryentryfield}[5]{%
  \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{superragged3colborder}{%
  \csuse{@glscmpstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheader}{%
  \csuse{@glscmpstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheaderborder}{%
  \csuse{@glscmpstyle@superragged3col}%
}%
\compatglossarystyle{altsuperragged4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%
\compatglossarystyle{altsuperragged4colheader}{%
  \csuse{@glscmpstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colborder}{%
  \csuse{@glscmpstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colheaderborder}{%
  \csuse{@glscmpstyle@altsuperragged4col}%
}%
\compatglossarystyle{super}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
}%
\compatglossarystyle{superborder}{%
  \csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{superheader}{%
  \csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{superheaderborder}{%
  \csuse{@glscmpstyle@super}%
}%

```

```

\compatglossarystyle{super3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\\}%
}%
\compatglossarystyle{super3colborder}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super3colheader}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super3colheaderborder}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
}%
\compatglossarystyle{super4colheader}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{super4colborder}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{super4colheaderborder}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4col}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colheader}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colborder}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colheaderborder}{%
  \csuse{@glscompstyle@super4col}%
}%

```

8.6 Rollback v4.46 (glossaries-polyglossia-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-polyglossia}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \PackageWarningNoLine{glossaries}%
      {No language module detected for ‘\this@dialect’.\MessageBreak
      Language modules need to be installed separately.\MessageBreak
      Please check on CTAN for a bundle called\MessageBreak
      ‘glossaries-\CurrentTrackedLanguage’ or similar}%
    }%
  }%
}%
{%
}
```

8.7 Rollback v4.46 (glossaries-prefix-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-prefix}[2020/03/19 v4.46 (NLCT)]
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
\ProcessOptions
\RequirePackage{glossaries}[=v4.46]
\define@key{glossentry}{prefixfirst}{\def\@glo@entryprefixfirst{#1}}%
\define@key{glossentry}{prefixfirstplural}{\def\@glo@entryprefixfirstplural{#1}}%
\define@key{glossentry}{prefix}{\def\@glo@entryprefix{#1}}%
\define@key{glossentry}{prefixplural}{\def\@glo@entryprefixplural{#1}}%
\appto\@gls@keymap{,%
  {prefixfirst}{prefixfirst},%
  {prefixfirstplural}{prefixfirstplural},%
  {prefix}{prefix},%
  {prefixplural}{prefixplural}%
}
\appto\@newglossaryentryprehook{%
  \def\@glo@entryprefix{}%
  \def\@glo@entryprefixplural{}%
  \let\@glo@entryprefixfirst\@gls@default@value
```

```

\let\@glo@entryprefixfirstplural\@gls@default@value
}
\appto\@newglossaryentryposthook{%
\gls@assign@field{}\@glo@label\{prefix\}\@glo@entryprefix}%
\gls@assign@field{}\@glo@label\{prefixplural\}\@glo@entryprefixplural}%
\expandafter\gls@assign@field\expandafter
\csname glo@\@glo@label @prefix\endcsname\{\@glo@label\{prefixfirst\}%
\{\@glo@entryprefixfirst\}%
\expandafter\gls@assign@field\expandafter
\csname glo@\@glo@label @prefixplural\endcsname\{\@glo@label\}%
\{prefixfirstplural\}\@glo@entryprefixfirstplural}%
}
\newcommand*\@glsentryprefixfirst}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixfirst}}
\newcommand*\@glsentryprefixfirstplural}[1]{%
\csuse{glo@\glsdetoklabel{#1}@prefixfirstplural}}
\newcommand*\@glsentryprefix}[1]{\csuse{glo@\glsdetoklabel{#1}@prefix}}
\newcommand*\@glsentryprefixplural}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixplural}}
\newrobustcmd*\@Glsentryprefixfirst}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixfirst\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\@Glsentryprefixfirstplural}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixfirstplural\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\@Glsentryprefix}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefix\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\@Glsentryprefixplural}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixplural\endcsname}%
\xmakefirstuc\@glo@text
}
\newcommand*\@ifglshasprefix}[3]{%
\ifcsempy{glo@\glsdetoklabel{#1}@prefix}%
{#3}%
{#2}%
}
\newcommand*\@ifglshasprefixplural}[3]{%
\ifcsempy{glo@\glsdetoklabel{#1}@prefixplural}%
{#3}%
{#2}%
}
\newcommand*\@ifglshasprefixfirst}[3]{%
\ifcsempy{glo@\glsdetoklabel{#1}@prefixfirst}%
{#3}%
{#2}%
}
\newcommand*\@ifglshasprefixfirstplural}[3]{%
\ifcsempy{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}%
    {%
      \ifgls@hasprefix{#2}{\glsentryprefix{#2}\glsprefixsep}{}%
    }%
    {%
      \ifgls@hasprefixfirst{#2}{\glsentryprefixfirst{#2}\glsprefixsep}{}%
    }%
    \@gls@{#1}{#2}[#3]%
  }%
}
\newrobustcmd{\pglsp1}{\@gls@hyp@opt\@pglsp1}
\newcommand*{\@pglsp1}[2][{}]{%
  \new@ifnextchar[%
    {\@pglsp1@{#1}{#2}}%
    {\@pglsp1@{#1}{#2}[]}%
}
\def\@pglsp1@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifgls@hasprefixplural{#2}{\glsentryprefixplural{#2}\glsprefixsep}{}%
    }%
    {%
      \ifgls@hasprefixfirstplural{#2}%
      {\glsentryprefixfirstplural{#2}\glsprefixsep}{}%
    }%
    \@glspl@{#1}{#2}[#3]%
  }%
}

```

```

\newrobustcmd{\Pgl}{\@gls@hyp@opt\Pgl}
\newcommand*{\@Pgl}[2][{}]{%
  \new@ifnextchar[%
    {\@Pgl@{#1}{#2}}%
    {\@Pgl@{#1}{#2}[]}%
}
\def\@Pgl@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglshasprefix{#2}%
      {%
        \Glsentryprefix{#2}%
        \glsprefixsep
        \@gls@{#1}{#2}[#3]%
      }%
      {\@Gls@{#1}{#2}[#3]}%
    }%
    {%
      \ifglshasprefixfirst{#2}%
      {%
        \Glsentryprefixfirst{#2}%
        \glsprefixsep
        \@gls@{#1}{#2}[#3]%
      }%
      {\@Gls@{#1}{#2}[#3]}%
    }%
  }%
}
\newrobustcmd{\Pglsp}{\@gls@hyp@opt\Pglsp}
\newcommand*{\@Pglsp}[2][{}]{%
  \new@ifnextchar[%
    {\@Pglsp@{#1}{#2}}%
    {\@Pglsp@{#1}{#2}[]}%
}
\def\@Pglsp@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglshasprefixplural{#2}%
      {%
        \Glsentryprefixplural{#2}%
        \glsprefixsep
        \@glsp@{#1}{#2}[#3]%
      }%
      {\@Glspl@{#1}{#2}[#3]}%
    }%
  }%
}

```

```

}%
{%
  \ifglshasprefixfirstplural{#2}%
  {%
    \Glsentryprefixfirstplural{#2}%
    \glsprefixsep
    \@glsp1@{#1}{#2}[#3]%
  }%
  {\@Glspl@{#1}{#2}[#3]}%
}%
}%
}
\newrobustcmd{\PGLS}{\@glshyp@opt\PGLS}
\newcommand*{\@PGLS}[2][{}]{%
  \new@ifnextchar[%
    {\@PGLS@{#1}{#2}}%
    {\@PGLS@{#1}{#2}[]}%
}
\def\@PGLS@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glstoifexists{#2}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefix{#2}%
      {\mfirstucMakeUppercase{\glentryprefix{#2}\glsprefixsep}}{%
    }%
    {%
      \ifglshasprefixfirst{#2}%
      {\mfirstucMakeUppercase{\glentryprefixfirst{#2}\glsprefixsep}}{%
    }%
    \@GLS@{#1}{#2}[#3]%
  }%
}
}
\newrobustcmd{\PGLSpl}{\@glshyp@opt\PGLSpl}
\newcommand*{\@PGLSpl}[2][{}]{%
  \new@ifnextchar[%
    {\@PGLSpl@{#1}{#2}}%
    {\@PGLSpl@{#1}{#2}[]}%
}
\def\@PGLSpl@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glstoifexists{#2}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefixplural{#2}%
      {\mfirstucMakeUppercase{\glentryprefixplural{#2}\glsprefixsep}}{%
    }%
    {%

```



```

\ifglshasprefixfirstplural{#2}%
{\mfirstucMakeUppercase{\glstryprefixfirstplural{#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@hypergroup@list@#1\endcsname
  \@for\@gls@elem:=\@gls@list\do{%
    \ifthenelse{\equal{\@gls@elem}{#2}}{\@endfortrue}{}%
  }
  \if@endfor
  \else
    \GlossariesWarningNoLine{Navigation panel
      for glossary type ‘#1’^^Jmissing group ‘#2’}%
    \gdef\gls@hypergroup@prerun{%
      \GlossariesWarningNoLine{Navigation panel
        has changed. Rerun LaTeX}%
    }
  \fi
}
\let\gls@hypergroup@prerun\relax
\AtEndDocument{\gls@hypergroup@prerun}
\newcommand*{\@gls@hypergroup}[2]{%
  \@ifundefined{\@gls@hypergroup@list@#1}{%
    \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{#2}%
  }{%
    \expandafter\let\expandafter\@gls@tmp
      \csname @gls@hypergroup@list@#1\endcsname
    \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{%
      \@gls@tmp,#2}%
  }%
}
\newcommand*{\glsnavigation}{%
  \def\@gls@between{}%
  \ifcsundef{\@gls@hypergroup@list@\@glo@type}%

```

```

{%
  \def\@gls@list{%
}%
{%
  \expandafter\let\expandafter\@gls@list
    \csname @gls@hypergroup@list@\@glo@type\endcsname
}%
\@for\@gls@tmp:=\@gls@list\do{%
  \@gls@between
  \@gls@getgrouptitle{\@gls@tmp}{\@gls@grptitle}%
  \glsnavhyperlink{\@gls@tmp}{\@gls@grptitle}%
  \let\@gls@between\glshypernavsep
}%
}
\newcommand*\@gls@hypernavsep{\space\textbar\space}
\newcommand*\@gls@symbolnav{%
  \glsnavhyperlink{glsymbols}{\@gls@getgrouptitle{glsymbols}}%
  \glshypernavsep
  \glsnavhyperlink{glsnumbers}{\@gls@getgrouptitle{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{%
    }%
    {\gls@postinline}%
  \renewcommand*\@glossaryheader{}%
  \renewcommand*\@gls@groupheading[1]{%
  \renewcommand*\@glossentry[2]{%
    \gls@inlinedopostchild
    \gls@inlinesep
    \gls@entryitem{##1}%
    \gls@inlinenameformat{##1}%
    \glossentryname{##1}%
  }%
  \ifglsdescsuppressed{##1}%
  {%
    \gls@inlineemptydescformat
    {%
      \glossentrysymbol{##1}%
    }%
  }%
}

```

```

    {%
      ##2%
    }%
  }%
  {%
    \ifglshasdesc{##1}%
    {\glsinlinedescformat{\glossentrydesc{##1}}{\glossentrysymbol{##1}}{##2}}%
    {\glsinlineemptydescformat{\glossentrysymbol{##1}}{##2}}%
  }%
  \ifglshaschildren{##1}%
  {%
    \glsresetsubentrycounter
    \glsinlineparentchildseparator
    \def\gls@inlinesubsep{}%
    \def\gls@inlinepostchild{\glsinlinepostchild}%
  }%
  {}%
  \def\gls@inlinesep{\glsinlineseparator}%
}%
\renewcommand{\subglossentry}[3]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{%
    \glossentryname{##2}}%
  \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[\glstryitem{##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{\glsgrouptitle{##1}}]}
\newglossarystyle{listhypergroup}{%
  \setglossarystyle{list}%
  \renewcommand*\glossaryheader{%
    \glslistnavigationitem{\glslnavigation}}%
  \renewcommand*\glsgroupheading[1]{%
    \item[\glslistgroupheaderfmt
      {\glslnavhypertarget{##1}{\glsgrouptitle{##1}}]}
\newglossarystyle{altlist}{%
  \setglossarystyle{list}%
  \renewcommand*\glossentry[2]{%
    \item[\glstryitem{##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{\glsgrouptitle{##1}}]}
\newglossarystyle{altlisthypergroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*\glossaryheader{%
    \glslistnavigationitem{\glslnavigation}}%

```

```

\renewcommand*{\glsgroupheading}[1]{%
  \item[\glslistgroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}]%
\newglossarystyle{listdotted}{%
  \setglossarystyle{list}%
  \renewcommand*{\glossentry}[2]{%
    \item[\makebox[\glslistdottedwidth][l]{%
      \glentryitem{##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[\glentryitem{##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]{%
    \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &

```

```

        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
        ##3\tabularnewline
    }%
    \ifglsgroupskip
        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
    \fi
}
\newglossarystyle{longborder}{%
    \setglossarystyle{long}%
    \renewenvironment{theglossary}{%
        \begin{longtable}{|l|p{\glstdescwidth}|}{\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{\glstdescwidth}p{\glspagelistwidth}}}%
        {\end{longtable}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupeheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
        \glssubentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1} & ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2} &
        ##3\tabularnewline
    }%
    \ifglsgroupskip
        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
    \fi

```

```

}
\newglossarystyle{long3colborder}{%
  \setglossarystyle{long3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|p{\glstdescwidth}|p{\glspagelistwidth}|}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long3colheader}{%
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long3colheaderborder}{%
  \setglossarystyle{long3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}
\newglossarystyle{long4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l|l|l|l}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupeheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{long4colheader}{%
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&

```

```

        \bfseries \symbolname&
        \bfseries\pagelistname\tabularnewline\endhead}%
    }
    \newglossarystyle{long4colborder}{%
        \setglossarystyle{long4col}%
        \renewenvironment{theglossary}%
            {\begin{longtable}{|l|l|l|l|}}%
            {\end{longtable}}%
        \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
    }
    \newglossarystyle{long4colheaderborder}{%
        \setglossarystyle{long4col}%
        \renewenvironment{theglossary}%
            {\begin{longtable}{|l|l|l|l|}}%
            {\end{longtable}}%
        \renewcommand*{\glossaryheader}{%
            \hline\bfseries\entryname&\bfseries\descriptionname&
            \bfseries \symbolname&
            \bfseries\pagelistname\tabularnewline\hline\endhead
            \hline\endfoot}%
    }
    \newglossarystyle{altlong4col}{%
        \setglossarystyle{long4col}%
        \renewenvironment{theglossary}%
            {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
            {\end{longtable}}%
    }
    \newglossarystyle{altlong4colheader}{%
        \setglossarystyle{long4colheader}%
        \renewenvironment{theglossary}%
            {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
            {\end{longtable}}%
    }
    \newglossarystyle{altlong4colborder}{%
        \setglossarystyle{long4colborder}%
        \renewenvironment{theglossary}%
            {\begin{longtable}{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%
            {\end{longtable}}%
    }
    \newglossarystyle{altlong4colheaderborder}{%
        \setglossarystyle{long4colheaderborder}%
        \renewenvironment{theglossary}%
            {\begin{longtable}{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%
            {\end{longtable}}%
    }
}

```

8.12 Rollback v4.46 (glossary-longbooktabs-2020-03-19.sty)

Version 4.46 preserved for rollback.


```

\ProvidesPackage{glossary-longbooktabs}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{booktabs}
\RequirePackage{glossary-long}[=v4.46]
\RequirePackage{glossary-longragged}[=v4.46]
\newglossarystyle{long-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname & \bfseries
    \descriptionname\tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{long3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname &
    \bfseries \descriptionname &
    \bfseries \pagelistname
    \tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{long4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname &
    \bfseries \descriptionname &
    \bfseries \symbolname &
    \bfseries \pagelistname
    \tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{altlong4col-booktabs}{%
  \glspatchLToutput

```

```

\setglossarystyle{long4col-booktabs}%
\renewenvironment{theglossary}%
  {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
  {\end{longtable}}%
}
\newglossarystyle{longragged-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{longragged3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlongragged4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{altlong4col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newcommand*{\glsLTpenaltycheck}{%
  \ifnum\outputpenalty=-50\vskip-\normalbaselineskip\relax\fi
}
\newcommand{\glspenaltygroupskip}{%
  \noalign{\penalty-50\vskip\normalbaselineskip}}
\let\@gls@org@LT@output\LT@output
\newcommand*{\glsrestoreLToutput}{\let\LT@output\@gls@org@LT@output}
\newcommand*{\glspatchLToutput}{%
  \renewcommand*{\LT@output}{%
    \ifnum\outputpenalty <-\@Mi
      \ifnum\outputpenalty > -\LT@end@pen
        \LT@err{floats and marginpars not allowed in a longtable}\@ehc
      \else
        \setbox\z@\vbox{\unvbox\@cclv}%
        \ifdim \ht\LT@lastfoot>\ht\LT@foot
          \dimen@ \pagegoal
          \advance\dimen@-\ht\LT@lastfoot
          \ifdim\dimen@<\ht\z@
            \setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%
            \@makecol
          \fi
          \@outputpage
        \setbox\z@\vbox{\box\LT@head\glsLTpenaltycheck}%
      \fi
    }%
  }%
}

```

```

\fi
\fi
\global\@colroom\@colht
\global\vsizel\@colht
{\unvbox\z@box\ifvoidLT@lastfootLT@footelseLT@lastfoot\fi}%
\fi
\else
\setbox\@cclv\box{\unvbox\@cclv\copyLT@foot\vss}%
\@makecol
\@outputpage
\global\vsizel\@colroom
\copyLT@head
\glstpenaltycheck
\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\hsizel}
}{%}
\@ifundefined{glspagelistwidth}{%
\newlength{glspagelistwidth}
\setlength{glspagelistwidth}{0.1\hsizel}
}{%}
\newglossarystyle{longragged}{%
\renewenvironment{theglossary}{%
{\begin{longtable}{l>{\raggedright}p{glsdescwidth}}}%
{\end{longtable}}}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1}\glspostdescription\space ##2%
\tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2}%
\glspostdescription\space ##3%
\tabularnewline
}

```

```

}%
\ifglsgroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
\fi
}
\newglossarystyle{longraggedborder}{%
  \setglossarystyle{longragged}%
  \renewenvironment{theglossary}{%
    \begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|}%
    {\end{longtable}}%
    \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
  }
\newglossarystyle{longraggedheader}{%
  \setglossarystyle{longragged}%
  \renewcommand*{\glossaryheader}{%
    \bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\endhead}%
  }
\newglossarystyle{longraggedheaderborder}{%
  \setglossarystyle{longraggedborder}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\hline
    \endhead
    \hline\endfoot}%
  }
\newglossarystyle{longragged3col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
      \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1} & ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
      &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2} &
      ##3\tabularnewline
    }%
  }
\ifglsgroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
\fi

```

```

}
\newglossarystyle{longragged3colborder}{%
  \setglossarystyle{longragged3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|}%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{longragged3colheader}{%
  \setglossarystyle{longragged3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{longragged3colheaderborder}{%
  \setglossarystyle{longragged3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}
\newglossarystyle{altlongragged4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
  \fi
}
\newglossarystyle{altlongragged4colheader}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%

```

```

        {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
          >{\raggedright}p{\glspagelistwidth}}}%
        {\end{longtable}}}%
\renewcommand*{\glossaryheader}{%
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{altlongragged4colborder}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|%
      >{\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{\glsgrouptitle{##1}}\indexspace}%
}
\newglossarystyle{mcolindexhypergroup}{%
\setglossarystyle{mcolindex}%
\renewcommand*{\glossaryheader}{%
\item\glstreenavigationfmt{\glsnavigation}\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgrouptitle{##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}{\glsgrouptitle{##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{\glsgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreehypergroup}{%
\setglossarystyle{mcoltree}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
}

```

```

\renewcommand*{\glsgroupeheading}[1]{%
  \par\noindent
  \glstreegroupheaderfmt{\glsnahypertarget{##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*{\glsgroupeheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnahypertarget{##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{\glsgroupeheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreenonamehypergroup}{%
  \setglossarystyle{mcoltreenoname}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupeheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnahypertarget{##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{mcolalanttreetree}{%
  \setglossarystyle{almtree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
}
\newglossarystyle{mcolalanttreetreegroup}{%
  \setglossarystyle{mcolalanttreetree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcolalanttreetreehypergroup}{%
  \setglossarystyle{mcolalanttreetree}%
  \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{mcolalanttreespannav}{%
  \setglossarystyle{almtree}%
  \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]{%
    \glstryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%

```

```

        \begin{supertabular}{|l|p{\glsdescwidth}|}%
        {\end{supertabular}}%
    }
    \newglossarystyle{superheader}{%
        \setglossarystyle{super}%
        \renewenvironment{theglossary}%
        {\tablehead{\bfseries \entryname &
            \bfseries\descriptionname\tabularnewline}%
            \tabletail{}}%
        {\begin{supertabular}{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
        }%
        \ifglsgroupskip
            \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{\glsgdescwidth}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{\glsgdescwidth}|p{\glspagelistwidth}|}%
      {\end{supertabular}}}%
    }%
}
\newglossarystyle{super4col}{%
  \renewenvironment{theglossary}%
    {\tablehead{}}\tabletail{}}%
    {\begin{supertabular}{llll}}}%
    {\end{supertabular}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
  \fi
}
\newglossarystyle{super4colheader}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}%
    {\tabletail{}}%

```

```

        \begin{supertabular}{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|l|}}%
            {\end{supertabular}}%
    }
    \newglossarystyle{altsuper4col}{%
        \setglossarystyle{super4col}%
        \renewenvironment{theglossary}%
            {\tablehead{} \tabletail{}}%
            {\begin{supertabular}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
            {\end{supertabular}}%
    }
    \newglossarystyle{altsuper4colheader}{%
        \setglossarystyle{super4colheader}%
        \renewenvironment{theglossary}%
            {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
                        \bfseries\symbolname &
                        \bfseries\pagelistname\tabularnewline}\tabletail{}}%
            {\begin{supertabular}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
            {\end{supertabular}}%
    }
    \newglossarystyle{altsuper4colborder}{%
        \setglossarystyle{super4colborder}%
        \renewenvironment{theglossary}%
            {\tablehead{\hline}\tabletail{\hline}%
             \begin{supertabular}%
                 {l|lp{\glsgdescwidth}|l|lp{\glspagelistwidth}|}}%
            {\end{supertabular}}%
    }
    \newglossarystyle{altsuper4colheaderborder}{%
        \setglossarystyle{super4colheaderborder}%
        \renewenvironment{theglossary}%
            {\tablehead{\hline
                        \bfseries\entryname &
                        \bfseries\descriptionname &

```

```

        \bfseries\symbolname &
        \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}%
        {1|p{\glsdescwidth}|1|p{\glspagelistwidth}|}%
    \end{supertabular}}%
}

```

8.16 Rollback v4.46 (glossary-superragged-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-superragged}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{array}
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
    \newlength{glsdescwidth}
    \setlength{glsdescwidth}{0.6\hsize}
}{%
}
\@ifundefined{glspagelistwidth}{%
    \newlength{glspagelistwidth}
    \setlength{glspagelistwidth}{0.1\hsize}
}{%
}
\newglossarystyle{superragged}{%
    \renewenvironment{theglossary}%
        {\tablehead{}\tabletail{}}%
        {\begin{supertabular}{1>\raggedright}p{\glsdescwidth}}}%
    {\end{supertabular}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
        \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1}\glspostdescription\space ##2%
        \tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
        ##3%
        \tabularnewline
    }%
    \ifglsnogroupskip
        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{& \tabularnewline}%
    \fi
}
\newglossarystyle{superraggedborder}{%
    \setglossarystyle{superragged}%
}

```

```

\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
   \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}}}%
  {\end{supertabular}}%
}
\newglossarystyle{superraggedheader}{%
  \setglossarystyle{superragged}%
\renewenvironment{theglossary}%
  {\tablehead{\bfseries \entryname & \bfseries \descriptionname
   \tabularnewline}%
   \tabletail{}}%
  {\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}}}%
  {\end{supertabular}}%
}
\newglossarystyle{superraggedheaderborder}{%
  \setglossarystyle{superragged}%
\renewenvironment{theglossary}%
  {\tablehead{\hline\bfseries \entryname &
   \bfseries \descriptionname\tabularnewline\hline}%
   \tabletail{\hline}%
   \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}}}%
  {\end{supertabular}}%
}
\newglossarystyle{superragged3col}{%
\renewenvironment{theglossary}%
  {\tablehead{}%
   \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*{\glsgruppeheading}[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{\glsgdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuperragged4colborder}{%
  \setglossarystyle{altsuper4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}%
      {ll>{\raggedright}p{\glsgdescwidth}ll%
      >{\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}%
      {ll>{\raggedright}p{\glsgdescwidth}ll%
      >{\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\glssentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
\ifglshassymbol{##1}{\space\glossentrysymbol{##1}}{}%
\glstreepredesc \glossentrydesc{##1}\glspostdescription\space ##2%
}%
\renewcommand{\subglossentry}[3]{%
\ifcase##1\relax
% level 0
\item
\or
% level 1
\subitem
\glssubentryitem{##2}%
\else
% all other levels
\subsubitem
\fi
\glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
\ifglshassymbol{##2}{\space\glossentrysymbol{##2}}{}%
\glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3%
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
\newglossarystyle{indexgroup}{%
\setglossarystyle{index}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt{\glsgrouptitle{##1}}%
\indexspace
}%
}
\newglossarystyle{indexhypergroup}{%
\setglossarystyle{index}%
\renewcommand*{\glossaryheader}{%

```

```

\item\glstreenavigationfmt{\glsnavigation}\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavigationhypertarget{##1}{\glsgrouptitle{##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
\glstentryitem{##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{\glsgrouptitle{##1}}\par
\indexspace}%
}
\newglossarystyle{treehypergroup}{%
\setglossarystyle{tree}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt
{\glsnavigationhypertarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}
\newlength\glstreeindent
\setlength{\glstreeindent}{10pt}
\newglossarystyle{treenoname}{%

```

```

\renewenvironment{theglossary}%
{
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}}%
{}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\hangindent0pt\relax
\parindent0pt\relax
\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}%
{%
\forallglsentries[\@gls@type]{\@glo@label}%
{%
\ifglshasparent{\@glo@label}%
{}}%
}%
}

```

```

\settowidth{\dimen@}{%
  {\glstreenamefmt{\glstentryname{\@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}{%
      \glstentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}}}%
    \ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}%
    \glossentrydesc{##1}\glspostdescription \space ##2\par
    \def\@gls@prevlevel{0}%
  }%
  \renewcommand{\subglossentry}[3]{%
    \ifnum##1=1\relax
      \glssubentryitem{##2}%
    \fi
    \ifnum\@gls@prevlevel=##1\relax
    \else
      \@ifundefined{@glswidestname\romannumeral##1}{%
        \settowidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname}\space}}{%
        \settowidth{\gls@tmplen}{\glstreenamefmt{%
          \csname @glswidestname\romannumeral##1\endcsname\space}}}%
      \ifnum\@gls@prevlevel<##1\relax
        \setlength{\glstreeindent}{\gls@tmplen

```

```

\addtolength\glstreeindent\parindent
\parindent\glstreeindent
\else
\@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
\settowidth{\glstreeindent}{\glstreenamefmt{%
\@glswidestname\space}}}%
\settowidth{\glstreeindent}{\glstreenamefmt{%
\csname @glswidestname\romannumeral\@gls@prevlevel
\endcsname\space}}}%
\addtolength\parindent{-\glstreeindent}%
\setlength\glstreeindent\parindent
\fi
\fi
\hangindent\glstreeindent
\makebox[Opt][r]{\glstreenamebox{\gls@tmplen}{%
\glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
\ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
\glossentrydesc{##2}\glspostdescription\space ##3\par
\def\@gls@prevlevel{##1}%
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{almtreegroup}{%
\setglossarystyle{almtree}%
\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
\par\indexspace}%
}
\newglossarystyle{almtreehypergroup}{%
\setglossarystyle{almtree}%
\renewcommand*{\glossaryheader}{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}}

```

8.18 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]{%
  \ifstrempy{#1}%
  {\@glsshowtarget{/#2 (#3)}}%
  {\@glsshowtarget{/#2 (#3) [#1]}}%
}
\newcommand*{\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries\space (or move
  \string\newglossaryentry\space
  definitions into the preamble)}}%
{You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key. It may
  be that the 'see' key has been written to the .glsdefs
  file from the previous run, in which case you need to
  move your definitions
  to the preamble if you don't want to use
  \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoinde}x{%
  [\gls@seenoinde@val\gls@seenoinde@nr]{error,warn,ignore}{%
    \ifcase\gls@seenoinde@nr
      \renewcommand*{\@gls@see@noindex}{%
        \PackageError{glossaries}%
        {'\gls@xr@key' key may only be used after \string\makeglossaries\space
        or \string\makenoidxglossaries}%
        {You must use \string\makeglossaries\space
        or \string\makenoidxglossaries\space before defining
        any entries that have a '\gls@xr@key' key}}%
      }%
    \or
      \renewcommand*{\@gls@see@noindex}{%
        \GlossariesWarning{'\gls@xr@key' key ignored}%
      }%
    \or
      \renewcommand*{\@gls@see@noindex}{}%
    \fi
  }
\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}%
  {\newcommand*{\@glossarysec}{section}}%
  {\newcommand*{\@glossarysec}{chapter}}
\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
  subsection,subsubsection,paragraph,subparagraph}[section]{%

```

```

\renewcommand*{\@glossarysec}{#1}}
\newcommand*{\@glossarysecstar}{*}%
\newcommand*{\@glossaryseclabel}{}%
\newcommand*{\glsautoprefix}{%
\define@choicekey{glossaries.sty}{numberedsection}%
[\gls@numberedsection@val\gls@numberedsection@nr]{%
false,nolabel,autolabel,nameref}[nolabel]{%
\ifcase\gls@numberedsection@nr\relax
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{}%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{}%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{%
\label{\glsautoprefix@glo@type}}%
\or
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix@glo@type}}%
\fi
}
\@ifpackageloaded{classicthesis}
{\newcommand*{\@glossary@default@style}{index}}
{\newcommand*{\@glossary@default@style}{list}}
\define@key{glossaries.sty}{style}{%
\def\@glossary@default@style{#1}%
}
\newcommand*{\@gls@declareoption}[2]{%
\DeclareOptionX{#1}{#2}%
\DeclareOption{#1}{#2}%
}
\newcommand*{\glossaryentrynumbers}[1]{#1\gls@save@numberlist{#1}}
\@gls@declareoption{nonumberlist}{%
\renewcommand*{\glossaryentrynumbers}[1]{\gls@save@numberlist{#1}}%
}
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{%
\gls@savenumberlistfalse
\newcommand*\@glo@seeautonumberlist{%
\@gls@declareoption{seeautonumberlist}{%
\renewcommand*\@glo@seeautonumberlist{%
\def\@glo@prefix{\glsnextpages}%
}%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{%
\gls@esclocationstrue
\newcommand*\@gls@loadlong{\RequirePackage{glossary-long}[=v4.49]}
\@gls@declareoption{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]{}
\glentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
  \renewcommand*{\@gls@counterwithin}{#1}%
  \glentrycountertrue
  \@gls@define@glossaryentrycounter
}
\newcommand*{\@gls@counterwithin}{}
\newcommand{\@gls@define@glossarysubentrycounter}{%
  \ifundef\c@glossarysubentry
  {%
    \ifglssubentrycounter
    \ifglentrycounter
      \newcounter{glossarysubentry}[glossaryentry]%
    \else
      \newcounter{glossarysubentry}%
    \fi
    \def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
  \fi
}%
}%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
\glssubentrycounterfalse
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none}{%
  \renewcommand*{\@glo@default@sorttype}{#1}%
  \csname @gls@setupsort@#1\endcsname
}
\newcommand*{\@gls@prestandardsort}[3]{%
  \glsdosanitizesort
}
\newcommand*{\@glo@check@sortallowed}[1]{}
\newcommand*{\@gls@setupsort@standard}{%
  \def\do@glo@storeentry{\@glo@storeentry}%
  \def\@gls@defsortcount##1{%
    \def\@gls@defsort##1##2{%
      \ifx\@glo@sort\@gls@defaultsort
        \let\@glo@sort\@glo@name
      \fi
      \let\glsdosanitizesort\@gls@sanitizesort
      \glsprestandardsort{\@glo@sort}{##1}{##2}%
      \expandafter\protected@xdef\csname glo###2@sort\endcsname{\@glo@sort}%
    }%
  \def\@gls@setsort##1{%
    \let\@glo@check@sortallowed\@gobble
  }
}
\@gls@setupsort@standard
\newcommand*\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 'theGLOSS' environment}%
}
\@gls@declareoption{noredefwarn}{%
  \renewcommand*{\@gls@warnonglossdefined}{}%
  \renewcommand*{\@gls@warnontheGLOSSdefined}{}%
}
\newcommand*{\@gls@sanitizedesc}{%
}
\newcommand*{\glssetexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@expand@field{##1}{#1}{##2}%
  }%
}
\newcommand*{\glssetnoexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@noexpand@field{##1}{#1}{##2}%
  }%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}
\glssetnoexpandfield{descplural}
\newcommand*{\@gls@sanitizename}{}
\glssetnoexpandfield{name}
\newcommand*{\@gls@sanitizesymbol}{}
\glssetnoexpandfield{symbol}
\glssetnoexpandfield{symbolplural}
\newcommand*{\@gls@sanitizesort}{%
  \ifgls@sanitizesort
    \@gls@sanitizesort
  \else
    \@gls@nosanitizesort
  \fi
}
\newcommand*{\@gls@sanitizesort}{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@gls@nosanitizesort}{}
\newcommand*{\@gls@noidx@sanitizesort}{%
  \ifdefvoid\@glo@sort
  {}%
  {%
    \expandafter\@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
  \def\@glo@sort{#1#2}%
}

```

```

\@onelevel@sanitize\@glo@sort
}
\newcommand*{\@@gls@noidx@nosanitizesort}{%
\ifdefvoid\@glo@sort
{}%
{%
\expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
}%
}
\def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
\bgroup
\glsnoidxstripaccents
\protected@xdef\@@glo@sort{#1#2}%
\egroup
\let\@glo@sort\@@glo@sort
}
\newcommand*\glsnoidxstripaccents{%
\let\IeC\@firstofone
\let\add@accent\@secondoftwo
\let\@text@composite\@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{nottranslate}{%
    \glstranslatefalse
    \let\@gls@usetranslator\relax
    \let\glsifusetranslator\@secondoftwo
}
\define@choicekey{glossaries.sty}{translate}{%
    [\gls@translate@val\gls@translate@nr]%
    {true,false,babel}[true]%
    {%
        \ifcase\gls@translate@nr\relax
            \glstranslatetrue
            \renewcommand*\@gls@usetranslator{%
                \@ifpackageloaded{polyglossia}%
                {%
                    \let\glsifusetranslator\@secondoftwo
                }%
                {%
                    \@ifpackageloaded{babel}%
                    {%
                        \IfFileExists{translator.sty}%
                        {%
                            \RequirePackage{translator}%
                            \let\glsifusetranslator\@firstoftwo
                        }%
                    }%
                }%
            }%
        }%
    }%
}
\or
    \glstranslatefalse
    \let\@gls@usetranslator\relax
    \let\glsifusetranslator\@secondoftwo
\or
    \glstranslatetrue
    \let\@gls@usetranslator\relax
    \let\glsifusetranslator\@secondoftwo
\fi
}
\glstranslatefalse

```

```

\let\glsifusetranslator\@secondoftwo
\@ifpackageloaded{translator}%
{%
  \glstranslatetrue
  \let\glsifusetranslator\@firstoftwo
}%
{%
  \@for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
  {
    \@ifpackageloaded{\gls@thissty}%
    {%
      \glstranslatetrue
      \@endfortrue
    }%
    {}%
  }
}
\define@boolkey{glossaries.sty}[gls]{indexonlyfirst}[true]{%
\glsindexonlyfirstfalse
\define@boolkey{glossaries.sty}[gls]{hyperfirst}[true]{%
\glshyperfirsttrue
\newcommand*{\@gls@setacrstyle}{%
\define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{%
\ifbool{glsacrdescription}%
{}%
}%
\renewcommand*{\@gls@sanitizedesc}{}%
}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{description}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{dua}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{shortcuts}[true]{%
\newcommand*{\glsorder}{word}
\newcommand*{\@glsorder}[1]{%
\define@choicekey{glossaries.sty}{order}{word,letter}{%

```

```

\def\glsorder{#1}}
\newif\ifglsxindy
\glsxindyfalse
\@gls@declareoption{makeindex}{\glsxindyfalse}
\define@boolkey[gls]{xindy}{glsnumbers}[true]{}
\gls@xindy@glsnumberstrue
\def\@xdy@main@language{\language}%
\define@key[gls]{xindy}{language}{\def\@xdy@main@language{#1}}
\ifcsundef{inputencodingname}{%
\def\gls@codepage{}}{%
\def\gls@codepage{\inputencodingname}
}
\define@key[gls]{xindy}{codepage}{\def\gls@codepage{#1}}
\define@key{glossaries.sty}{xindy}[]{%
\glsxindytrue
\setkeys[gls]{xindy}{#1}%
}
\@gls@declareoption{xindygloss}{%
\glsxindytrue
}
\@gls@declareoption{xindynoglsnumbers}{%
\glsxindytrue
\gls@xindy@glsnumbersfalse
}
\providecommand{\@domakeglossaries}[1]{#1}
\@gls@declareoption{disablemakegloss}{%
\ifdefequal\makeglossaries\@no@makeglossaries
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect
(\string\makenoidxglossaries\space has already been used)}}%
}%
{%
\ifdefequal\@makeglossary\@gobble
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect
(\string\makeglossaries\space has already been used)}}%
}%
{%
\renewcommand{\@domakeglossaries}[1]{%
\PackageInfo{glossaries}{\string\makeglossaries\space and
\string\makenoidxglossaries\space have been disabled}%
}%
}%
}%
}
\@gls@declareoption{restoremakegloss}{%
\ifdefequal\makeglossaries\@no@makeglossaries
{%
\GlossariesWarning{Option ‘restoremakegloss’ has no effect
(\string\makenoidxglossaries\space has already been used)}}%
}

```

```

}%
{%
  \ifdefequal\@makeglossary\@gobble
  {%
    \GlossariesWarning{Option 'restoremakegloss' has no effect
      (\string\makeglossaries\space has already been used)}%
  }%
  {%
    \PackageInfo{glossaries}{\string\makeglossaries\space and
      \string\makenoidxglossaries\space have been restored}%
    \let\@domakeglossaries\@firstofone
  }%
}%
}
\newcommand*{\@do@write@glslabels}{%
  \AtEndDocument{\@do@write@glslabels}%
  \let\@do@write@glslabels\relax
}
\newcommand*{\@do@write@glslabels}{%
  \newwrite\@glsl@labelsfile
  \immediate\openout\@glsl@labelsfile=\jobname.glslabels
  \protected@edef\@glsl@fulllist{\@glo@types}%
  \ifdefempty{\@ignored@glossaries}{}%
  {\protected@eappto\@glsl@fulllist{\@ignored@glossaries}}%
  \forallglseentries[\@glsl@fulllist]{\@glseentry}%
  {%
    \ifdefempty{\@glseentry}{}%
    {%
      \ifx\do@write@glslabelsname\@gobble
        \def\@glsl@thisentryname{%
      \else
        \letcs{\@glsl@thisentryname}{glo\@glsdetoklabel{\@glseentry}@name}%
        \@onelevel@sanitize\@glsl@thisentryname
      \fi
      \immediate\write\@glsl@labelsfile
        {\@glseentry\do@write@glslabelsname{\@glsl@thisentryname}}%
    }%
  }%
  \immediate\closeout\@glsl@labelsfile
}
\newcommand*{\do@write@glslabelsname}[1]{%
  \begingroup
  \catcode'\^^I12
  \gdef\@write@glslabelsname#1{\^^I#1}
  \endgroup
  \@glsl@declareoption{writeglslabels}{\@do@write@glslabels
    \let\do@write@glslabelsname\@gobble}
  \@glsl@declareoption{writeglslabelnames}{\@do@write@glslabels
    \let\do@write@glslabelsname\@write@glslabelsname}
  \newif\ifglslautomake

```



```

\newcommand{\gls@automake@nr}{1}
\define@choicekey{glossaries.sty}{automake}%
  [\gls@automake@val\gls@automake@nr]{true,false,immediate}[true]{%
  \ifnum\gls@automake@nr=1\relax
    \glsautomakefalse
  \else
    \glsautomaketrue
  \fi
  \ifglsautomake
    \renewcommand*{\@gls@doautomake}{%
      \PackageError{glossaries}{You must use
        \string\makeglossaries\space with automake=true}
        {%
          Either remove the automake=true setting or
          add \string\makeglossaries\space to your document preamble.%
        }%
    }%
  \else
    \renewcommand*{\@gls@doautomake}{}%
  \fi
}
\glsautomakefalse
\newcommand*{\@gls@doautomake}{}
\AtEndDocument{\@gls@doautomake}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
  \ifgls savewrites
    \renewcommand*{\glswritefiles}{\@glswritefiles}%
  \else
    \let\glswritefiles\@empty
  \fi
}
\gls savewritesfalse
\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@option
\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}{\glsup{t}\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]{%
    \glsnoxywarning\glsSetAlphaCompositor}
\fi
\@onlypremakeg\glsSetAlphaCompositor
\newcommand*{\gls@suffixF}{}
\newcommand*{\glsSetSuffixF}[1]{%
  \renewcommand*{\gls@suffixF}{#1}}
\@onlypremakeg\glsSetSuffixF
\newcommand*{\gls@suffixFF}{}
\newcommand*{\glsSetSuffixFF}[1]{%
  \renewcommand*{\gls@suffixFF}{#1}}

```

```

}
\ifcsundef{hyperlink}%
{%
  \newcommand*{\glsnumberformat}[1]{#1}%
}%
{%
  \newcommand*{\glsnumberformat}[1]{\glsnumber{#1}}%
}
\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*{\glossarypreamble}{%
  \csuse{@glossarypreamble@\currentglossary}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaulttype]{%
  \ifglossaryexists*{#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*{\glossarypostamble}{}
\newcommand*{\glossarysection}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@p@glossarysection{#1}{#2}%
  }%
  \gls@glossarymark{\glossarytoctitle}%
}
\ifcsundef{glossarymark}%
{%
  \newcommand{\gls@glossarymark}[1]{\glossarymark{#1}}
}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand{\gls@glossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUChead{#1}}{\memUChead{#1}}%
      \else
        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand{\gls@glossarymark}[1]{%
      \ifglsucmark
        \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
      \else

```



```

        \@mkboth{#1}{#1}%
    \fi
}
}
\providecommand{\glossarymark}[1]{%
    \ifglsucmark
        \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
    \else
        \@mkboth{#1}{#1}%
    \fi
}
\newcommand*{\setglossarysection}[1]{%
\setkeys{glossaries.sty}{section=#1}}
\newcommand*{\@glossarysection}[2]{%
    \ifdefempty\@glossarysecstar
    {%
        \csname\@glossarysec\endcsname[#1]{#2}%
    }%
    {%
        \csname\@glossarysec\endcsname*{#2}%
        \@gls@toc{#1}{\@glossarysec}%
    }%
    \@glossaryseclabel
}
\newcommand*{\@p@glossarysection}[2]{%
    \glsclearpage
    \phantomsection
    \ifdefempty\@glossarysecstar
    {%
        \csname\@glossarysec\endcsname{#2}%
    }%
    {%
        \@gls@toc{#1}{\@glossarysec}%
        \csname\@glossarysec\endcsname*{#2}%
    }%
    \@glossaryseclabel
}
\newcommand*{\gls@docclearpage}{%
    \ifthenelse{\equal{\@glossarysec}{chapter}}{%
    {%
        \ifcsundef{cleardoublepage}%
        {%
            \clearpage
        }%
        {%
            \ifcsdef{if@openright}%
            {%
                \if@openright
                \cleardoublepage
            }%
        }%
    }%
    }%
}

```

```

\else
\clearpage
\fi
}%
{%
\cleardoublepage
}%
}%
}%
{%
}%
}
\newcommand*{\glsclearpage}{\gls@docclearpage}
\newcommand*{\@gls@toc}[2]{%
\ifglstoc
\ifglsnumberline
\addcontentsline{toc}{#2}{\protect\numberline{#1}}%
\else
\addcontentsline{toc}{#2}{#1}%
\fi
\fi
}
\newcommand*{\glsnoindywarning}[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@gl saddx dy counters
  }
  \@onlypremakeg\GlsAddXdyAttribute
\else
  \newcommand*\GlsAddXdyAttribute[1]{%
    \gl snoxindywarning\GlsAddXdyAttribute}
\fi
\ifgl sxindy
\newcommand*{\@gl s@addpredefinedattributes}{%
  \GlsAddXdyAttribute{gl snumberformat}
  \GlsAddXdyAttribute{textrm}
  \GlsAddXdyAttribute{textsf}
  \GlsAddXdyAttribute{texttt}
  \GlsAddXdyAttribute{textbf}
  \GlsAddXdyAttribute{textmd}
  \GlsAddXdyAttribute{textit}
  \GlsAddXdyAttribute{textup}
  \GlsAddXdyAttribute{textsl}
  \GlsAddXdyAttribute{textsc}
  \GlsAddXdyAttribute{emph}
  \GlsAddXdyAttribute{gl shypernumber}
  \GlsAddXdyAttribute{hyper rm}
  \GlsAddXdyAttribute{hyper sf}
  \GlsAddXdyAttribute{hyper tt}
  \GlsAddXdyAttribute{hyper bf}
  \GlsAddXdyAttribute{hyper md}
  \GlsAddXdyAttribute{hyper it}
  \GlsAddXdyAttribute{hyper up}
  \GlsAddXdyAttribute{hyper sl}
  \GlsAddXdyAttribute{hyper sc}
  \GlsAddXdyAttribute{hyper emph}
  \GlsAddXdyAttribute{gl signore}
}
\else
  \let\@gl s@addpredefinedattributes\relax
\fi
\def\@xdyuseralphabets{}
\ifgl sxindy
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \edef\@xdyuseralphabets{%
      \@xdyuseralphabets ~^J
      (define-alphabet "#1" (#2))}}
\else
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \gl snoxindywarning\GlsAddXdyAlphabet}
\fi
\ifgl sxindy
  \newcommand*{\@gl s@xdy@locationlist}{%
    roman-page-numbers,%

```

```

    Roman-page-numbers,%
    arabic-page-numbers,%
    alpha-page-numbers,%
    Alpha-page-numbers,%
    Appendix-page-numbers,%
    arabic-section-numbers%
}
\protected@edef\@gls@roman{\@roman{0}\string"
    \string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@gls@roman
\edef\@gls@tmp{\string" \string"roman-numbers-lowercase\string"
    :sep \string"}}%
\@onelevel@sanitize\@gls@tmp
\ifx\@gls@tmp\@gls@roman
    \expandafter
        \edef\csname @gls@xdy\Lclass@roman-page-numbers\endcsname{%
            \string"roman-numbers-lowercase\string"%
        }%
\else
    \expandafter
        \edef\csname @gls@xdy\Lclass@roman-page-numbers\endcsname{%
            :sep \string"\@gls@roman\string"%
        }%
\fi
\expandafter\def\csname @gls@xdy\Lclass@Roman-page-numbers\endcsname{%
    \string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy\Lclass@arabic-page-numbers\endcsname{%
    \string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy\Lclass@alpha-page-numbers\endcsname{%
    \string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy\Lclass@Alpha-page-numbers\endcsname{%
    \string"ALPHA\string"%
}%
\expandafter\def\csname @gls@xdy\Lclass@Appendix-page-numbers\endcsname{%
    \string"ALPHA\string"
    :sep \string"\@gls@alphacompositor\string"
    \string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy\Lclass@arabic-section-numbers\endcsname{%
    \string"arabic-numbers\string"
    :sep \string"\glscompositor\string"
    \string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglxsindy

```

```

\newcommand\@xdycrossrefhook{}
\fi
\ifglxsindy
\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"{}\glssopenbrace\string" #3
:sep \string"\glsclosebrace\string"))
}%
\else
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep "\glssopenbrace"
#1
:sep "\glsclosebrace\glssopenbrace" #3
:sep "\glsclosebrace"))
}%
\fi
\edef\@xdyuserlocationnames{%
\@xdyuserlocationnames^^J\space\space\space
\string"#2\string"}%
}
\@onlypremakeg\GlsAddXdyLocation
\else
\newcommand*\GlsAddXdyLocation}[2]{%
\glsnosindywarning\GlsAddXdyLocation}
\fi
\ifglxsindy
\def\@xdylocationclassorder{^^J\space\space\space
\string"roman-page-numbers\string"^^J\space\space\space
\string"arabic-page-numbers\string"^^J\space\space\space
\string"arabic-section-numbers\string"^^J\space\space\space
\string"alpha-page-numbers\string"^^J\space\space\space
\string"Roman-page-numbers\string"^^J\space\space\space
\string"Alpha-page-numbers\string"^^J\space\space\space
\string"Appendix-page-numbers\string"
\@xdyuserlocationnames^^J\space\space\space
\string"see\string"
}
\fi
\ifglxsindy
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\def\@xdylocationclassorder{#1}}
\else
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\glsnosindywarning\GlsSetXdyLocationClassOrder}

```

```

\fi
\ifglxindy
  \def\@xdysortrules{}
\fi
\ifglxindy
  \newcommand*\GlsAddSortRule[2]{%
    \expandafter\toks@\expandafter{\@xdysortrules}%
    \protected@edef\@xdysortrules{\the\toks@ ^^J
      (sort-rule \string"#1\string" \string"#2\string"))}%
  }
\else
  \newcommand*\GlsAddSortRule[2]{%
    \glsnxindywarning\GlsAddSortRule}
\fi
\ifglxindy
  \def\@xdyrequiredstyles{tex}
\fi
\ifglxindy
  \newcommand*\GlsAddXdyStyle[1]{%
    \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
  \newcommand*\GlsAddXdyStyle[1]{%
    \glsnxindywarning\GlsAddXdyStyle}
\fi
\ifglxindy
  \newcommand*\GlsSetXdyStyles[1]{%
    \edef\@xdyrequiredstyles{#1}}
\else
  \newcommand*\GlsSetXdyStyles[1]{%
    \glsnxindywarning\GlsSetXdyStyles}
\fi
\newcommand*\findrootlanguage{-}
\def\@xdylanguage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
    \ifglossaryexists{#1}{%
      \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
    }{%
      \PackageError{glossaries}{Can't set language type for
        glossary type '#1' --- no such glossary}{%
        You have specified a glossary type that doesn't exist}}
\else
  \newcommand*\GlsSetXdyLanguage[2][]{%
    \glsnxindywarning\GlsSetXdyLanguage}
\fi
\def\@gls@codepage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyCodePage[1]{%
    \renewcommand*\@gls@codepage{#1}%
  }

```

```

\AtBeginDocument{%
  \ifx\gls@codepage\empty
    \ifpackageloaded{fontspec}{\def\gls@codepage{utf8}}{}%
  \fi
}
\else
  \newcommand*\GlsSetXdyCodePage[1]{%
    \glsnoxywarning\GlsSetXdyCodePage}
\fi
\ifglxindy
  \ifglxindy@glxnumbers
    \def\@xdylettergroups{(define-letter-group
      \string\glxnumbers\string^^J\space\space\space
      :prefixes (\string"0\string" \string"1\string"
        \string"2\string" \string"3\string" \string"4\string"
        \string"5\string" \string"6\string" \string"7\string"
        \string"8\string" \string"9\string")^^J\space\space\space
      \@xdynumbergrouporder)}
    \else
      \def\@xdylettergroups{}
    \fi
\fi
\newcommand*\GlsAddLetterGroup[2]{%
  \expandafter\toks@\expandafter{\@xdylettergroups}%
  \protected@edef\@xdylettergroups{\the\toks@^^J%
    (define-letter-group \string"#1\string"^^J\space\space\space#2)}}%
}%
\newcommand*\forallglossaries[3][\@glo@types]{%
  \@for#2:=#1\do{\ifx#2\empty\else#3\fi}%
}

\newcommand*\forallacronyms[2]{%
  \@for#1:=\@glsacronymlists\do{\ifx#1\empty\else#2\fi}%
}
\newcommand*\forallglsentries[3][\glsdefaulttype]{%
  \protected@edef\@glo@list{\csname glo@list@#1\endcsname}%
  \@for#2:=\@glo@list\do
  {%
    \ifdefempty{#2}{\fi}%
  }%
}
\newcommand*\forallglsentries[3][\@glo@types]{%
  \expandafter\forallglossaries\expandafter[#1]{\@this@glo@}%
  {%
    \forallglsentries[\@this@glo@]{#2}{#3}%
  }%
}
\newcommand*\ifglossaryexists{%
  \@ifstar\s@ifglossaryexists\@ifglossaryexists
}

```



```

\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{@glotype@#1@out}{#3}{#2}%
}
\newcommand{\s@ifglossaryexists}[3]{%
  \ifcsundef{glolist@#1}{#3}{#2}%
}
\newcommand*{\glsdetoklabel}[1]{#1}
\newcommand{\ifglsentryexists}[3]{%
  \ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}
\newcommand*{\ifglsused}[3]{%
  \ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}
\newcommand{\glsdoifexists}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
}
\newcommand{\glsdoifnoexists}[2]{%
  \ifglsentryexists{#1}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
      been defined}{}}{#2}%
}
\newcommand{\glsdoifexistsorwarn}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}%
  }%
}
\newcommand{\glsdoifexistsordo}[3]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}%
    #3%
  }%
}
\newcommand{\doifglossarynoexistsordo}[3]{%
  \ifglossaryexists*{#1}%
  {%
    \PackageError{glossaries}{Glossary type ‘#1’ already exists}{}%
    #3%
  }%
  {#2}%
}
\newrobustcmd{\ifglshaschildren}[3]{%
  \glsdoifexists{#1}%
  {%
    \def\do@glshaschildren{#3}%

```

```

\protected@edef\@gls@thislabel{\glsdetoklabel{#1}}%
\expandafter\forglentries\expandafter
  [\csname glo@\@gls@thislabel @type\endcsname]
{\glo@label}%
{%
  \letcs\glo@parent{glo@\glo@label @parent}%
  \ifdefequal\@gls@thislabel\glo@parent
  {%
    \def\do@glshaschildren{#2}%
    \@endfortrue
  }%
  {}%
}%
\do@glshaschildren
}%
}
\newcommand{\ifglshasparent}[3]{%
  \glsdoifexists{#1}%
  {%
    \ifcseempty{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
  }%
}
\newcommand*{\ifglshasdesc}[3]{%
  \ifcseempty{glo@\glsdetoklabel{#1}@desc}%
  {#3}%
  {#2}%
}
\newcommand*{\ifglshasdescsuppressed}[3]{%
  \ifcsequal{glo@\glsdetoklabel{#1}@desc}{\no@post@desc}%
  {#2}%
  {#3}%
}
\newrobustcmd*{\ifglshassymbol}[3]{%
  \letcs{\@glo@symbol}{glo@\glsdetoklabel{#1}@symbol}%
  \ifdefempty\@glo@symbol
  {#3}%
  {%
    \ifdefequal\@glo@symbol\@gls@default@value
    {#3}%
    {#2}%
  }%
}
\newrobustcmd*{\ifglshaslong}[3]{%
  \letcs{\@glo@long}{glo@\glsdetoklabel{#1}@long}%
  \ifdefempty\@glo@long
  {#3}%
  {%
    \ifdefequal\@glo@long\@gls@default@value
    {#3}%
    {#2}%
  }%
}

```

```

    }%
}
\newrobustcmd*{\ifglshasshort}[3]{%
  \letcs{\@glo@short}{glo\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}@#1}%
          \ifdefempty\@glo@thisvalue
            {%
              #4%
            }%
          {%
            \ifdefequal\@glo@thisvalue\@gls@default@value
              {%
                #4%
              }%
            {%
              \let\glscurrentfieldvalue\@glo@thisvalue
              #3%
            }%
          }%
        }%
      }%
    }%
  }%

```

```

    }%
  }%
}
{
  \GlossariesWarning{Unknown entry field ‘#1’}%
  #4%
}
}%
}%
}
}
\newcommand*{\glscurrentfieldvalue}{}
\newcommand*{@glo@types}{{},}
\newcommand*\@gls@provide@newglossary{%
  \protected@write\@auxout{}{\string\providecommand\string\@newglossary[4]{}}%
  \let\@gls@provide@newglossary\relax
}
\newcommand*{\defglsentryfmt}[2][\glsdefaultttype]{%
  \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\glsdefaultttype
    {%
      \gdef\glsdefaultttype{#2}%
    }{}%
    \toks@{#2}\protected@edef\@glo@types{\@glo@types\the\toks@,}%
    \expandafter\gdef\csname glolist@#2\endcsname{,}%
    \expandafter\edef\csname @gloptype@#2@log\endcsname{#1}%
    \expandafter\edef\csname @gloptype@#2@in\endcsname{#3}%
  }

```

```

\expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
\expandafter\@gls@forbidtextext\csname @glotype@#2@log\endcsname
\expandafter\@gls@forbidtextext\csname @glotype@#2@in\endcsname
\expandafter\@gls@forbidtextext\csname @glotype@#2@out\endcsname
\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}%
\ifglxsindy
\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*{\@gls@name}{%
  \PackageError{glossaries}{name key required in
    \string\newglossaryentry\space for entry '\@glo@label'}{You
    haven't specified the entry name}}

```



```

\newcommand*\@glsnodedesc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}%
\newcommand*\@glsdefaultplural{}
\newcommand*\@gls@missingnumberlist}[1]{%
  ??%
  \ifglssavenumberlist
    \GlossariesWarning{Missing number list for entry '#1'.
      Maybe makeglossaries + rerun required}%
  \else
    \PackageError{glossaries}%
    {Package option 'savenumberlist=true' required}%
    {%
      You must use the 'savenumberlist' package option
      to reference location lists.%
    }%
  \fi
}
\newcommand*\@glsdefaultsort{}\@glo@name
\newcount\gls@level
\newcommand*\@gls@noexpand@field}[3]{%
  \expandafter\global\expandafter
  \let\csname glo@#1@#2\endcsname#3%
}
\newcommand*\@gls@noexpand@fields}[4]{%
  \ifcsdef{gls@assign@#3@field}
  {%
    \ifdefequal{#4}\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \csuse{gls@assign@#3@field}{#2}\@gls@value}%
    }%
    {%
      \csuse{gls@assign@#3@field}{#2}{#4}%
    }%
  }%
  {%
    \ifdefequal{#4}\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \@gls@noexpand@field{#2}{#3}\@gls@value}%
    }%
    {%
  }

```

```

        \@gls@noexpand@field{#2}{#3}{#4}%
    }%
}%
}
\newcommand{\@gls@expand@field}[3]{%
\expandafter
\protected@xdef\csname glo@#1@#2\endcsname{#3}%
}
\newcommand{\@gls@expand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
{%
\@gls@expand@field{#2}{#3}{#4}%
}%
{%
\csuse{gls@assign@#3@field}{#2}{#4}%
}%
}%
}%
}%
\ifdefequal{#4}{\@gls@default@value}%
{%
\@gls@expand@field{#2}{#3}{#1}%
}%
{%
\@gls@expand@field{#2}{#3}{#4}%
}%
}%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
\def\@gls@tmp{#1}%
\ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*\{gls@expandfields}{%
\let\gls@assign@field\@gls@expand@fields
}
\newcommand*\{gls@noexpandfields}{%
\let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
\glsdoifnoexists{#1}%

```

```

    {%
      \gls@defglossaryentry{#1}{#2}%
    }%
  }
  \newcommand*{\gls@defdocnewglossaryentry}{%
    \let\newglossaryentry\new@glossaryentry
  }
  \newrobustcmd{\provideglossaryentry}[2]{%
    \ifglsentryexists{#1}%
    {%
      \gls@defglossaryentry{#1}{#2}%
    }%
  }
  \@onlypreamble{\provideglossaryentry}
  \newcommand*{\gls@glsdefs@labellist}{}
  \newcommand*{\gls@glsdefs@addlabel}[1]{\listxadd{\gls@glsdefs@labellist}{#1}}
  \newcommand*{\gls@docdef@labellist}{}
  \newcommand*{\gls@docdef@addlabel}[1]{\listxadd{\gls@docdef@labellist}{#1}}
  \newrobustcmd{\new@glossaryentry}[2]{%
    \ifundef\@gls@deffile
    {%
      \global\newwrite\@gls@deffile
      \immediate\openout\@gls@deffile=\jobname.glsdefs
    }%
    {%
      \ifglsentryexists{#1}%
      {%
        \xifinlist{#1}{\gls@docdef@labellist}%
        {%
          \GlossariesWarning{Entry ‘#1’ has already been defined in the
            document environment (ignoring new definition)}%
        }%
        {%
          \xifinlist{#1}{\gls@glsdefs@labellist}%
          {%
            \gls@defglossaryentry{#1}{#2}%
            \gls@docdef@addlabel{#1}%
            \@gls@writedef{#1}%
          }%
          {%
            \GlossariesWarning{Entry ‘#1’ has already been defined in
              the preamble (in which case remove the duplicate definition)
              or you have just updated to glossaries.sty
              v4.47 (in which case delete ‘\jobname.glsdefs’)}%
          }%
        }%
      }%
    }%
    \gls@defglossaryentry{#1}{#2}%
  }

```

```

\gls@docdef@addlabel{#1}%
\@gls@writedef{#1}%
}%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*\gls@begindocdefs{%
\@gls@enablesavenonumberlist
\edef\@gls@restoreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
\makeatletter
\InputIfFileExists{\jobname.glsdefs}{\gls@restoreat}{\gls@restoreat}
\undef\@gls@restoreat
\gls@defdocnewglossaryentry
}
\newcommand*\gls@defdocnewglossaryentry[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@docdef@addlabel{#1}%
\gls@defdocnewglossaryentry{#1}{#2}%
}%
}
\newcommand*\gls@writedef[1]{%
\immediate\write\@gls@deffile
{%
\string\gls@defdocnewglossaryentry{\gls@detoklabel{#1}}\glspercentchar^^J%
\expandafter\@gobble\string\{\glspercentchar%
}%
\@for\@gls@map:=\@gls@keymap\do
{%
\letcs\gls@value{gls@detoklabel{#1}}\expandafter\@secondoftwo\@gls@map}%
\ifdef\gls@value
{%
\@onelevel@sanitize\gls@value
\immediate\write\@gls@deffile
{%
\expandafter\@firstoftwo\@gls@map
=\expandafter\@gobble\string\{\gls@value\expandafter\@gobble\string\},%
\glspercentchar
}%
}%
}%
}
\gls@writedefhook
\immediate\write\@gls@deffile
{%
\glspercentchar^^J%
}

```

```

\expandafter\@gobble\string\}\glpercentchar^^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{\glsfieldxddef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \protected@csxdef{glo@\@glo@label @#2}{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\glsfielddedef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \protected@csedef{glo@\@glo@label @#2}{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\glsfieldgdef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\glsfieldddef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
    }%
  }%
}

```



```

}%
{%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
}%
}%
}
\newcommand{\glsfieldfetch}[3]{%
    \glsdoifexists{#1}%
    {%
        \protected@edef\@glo@label{\glsdetoklabel{#1}}%
        \ifcsdef{glo@\@glo@label @#2}%
        {%
            \letcs#3{glo@\@glo@label @#2}%
        }%
        {%
            \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
        }%
    }%
}
\newcommand*{\glsunexpandedfieldvalue}[2]{%
    \ifcsname glo@\glsdetoklabel{#1}@#2\endcsname
    \expandafter\expandonce\csname glo@\glsdetoklabel{#1}@#2\expandafter\endcsname
    \fi
}
\newcommand{\ifglsfieldeq}[5]{%
    \glsdoifexists{#1}%
    {%
        \protected@edef\@glo@label{\glsdetoklabel{#1}}%
        \ifcsdef{glo@\@glo@label @#2}%
        {%
            \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
        }%
        {%
            \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
        }%
    }%
}
\newcommand{\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\@gls@name
\let\@glo@desc\@gls@desc
\let\@glo@descplural\@gls@descplural
\let\@glo@type\@gls@type

```

```

\let\@glo@symbol\@gls@default@value
\let\@glo@symbolplural\@gls@default@value
\let\@glo@text\@gls@default@value
\let\@glo@plural\@gls@default@value
\let\@glo@first\@gls@default@value
\let\@glo@firstplural\@gls@default@value
\let\@glo@sort\@gls@default@value
\let\@glo@counter\@gls@default@value
\def\@glo@see{}%
\def\@glo@parent{}%
\def\@glo@prefix{}%
\@gls@initnonumberlist
\def\@glo@useri{}%
\def\@glo@userii{}%
\def\@glo@useriii{}%
\def\@glo@useriv{}%
\def\@glo@userv{}%
\def\@glo@uservi{}%
\def\@glo@short{}%
\def\@glo@shortpl{}%
\def\@glo@long{}%
\def\@glo@longpl{}%
\@newglossaryentryprehook
\setkeys{glossentry}{#2}%
\ifundef\glsdefaulttype
{%
  \PackageError{glossaries}%
  {No default glossary type (have you used ‘nomain’ by mistake?)}%
  {If you use package option ‘nomain’ you must define
   a new glossary before you can define entries}%
}%
}%
\gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
\protected@edef\@glo@type{\glsentrytype{\@glo@label}}%
\ifcsundef{glo@list@\@glo@type}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘\@glo@type’ has not been defined}%
  {You need to define a new glossary type, before making entries
   in it}%
}%
}%
\ifignoredglossary\@glo@type
{%
  \ifx\@glo@desc\@gls@nodesc
    \let\@glo@desc\@empty
  \fi
}%
}%
}%

```

```

\protected@edef\@glo@list@{\csname glo@list@\@glo@type\endcsname}%
\expandafter\protected@xdef\csname glo@list@\@glo@type\endcsname{%
  \@glo@list@{\@glo@label},}%
}%
\gls@level=0\relax
\ifx\@glo@parent\@empty
  \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
\else
  \ifdefequal\@glo@label\@glo@parent%
  {%
    \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
    \def\@glo@parent{}%
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
  }%
  {%
    \ifglsentryexists{\@glo@parent}%
    {%
      \expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
        \@glo@parent}%
      \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
      \advance\gls@level by 1\relax
      \ifx\@glo@name\@glsnoname
        \expandafter\let\expandafter\@glo@name
        \csname glo@\@glo@parent @name\endcsname
      \ifx\@glo@plural\@gls@default@value
        \expandafter\let\expandafter\@glo@plural
        \csname glo@\@glo@parent @plural\endcsname
      \fi
    \fi
  }%
  {%
    \PackageError{glossaries}%
    {%
      Invalid parent ‘\@glo@parent’
      for entry ‘\@glo@label’ - parent doesn’t exist%
    }%
    {%
      Parent entries must be defined before their children%
    }%
    \def\@glo@parent{}%
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
  }%
}%
\fi
\expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%
\gls@assign@field{\@glo@name}{\@glo@label}{\sortvalue}{\@glo@sort}%
\letcs\@glo@sort{glo@\@glo@label @sortvalue}%
\gls@assign@field{\@glo@name}{\@glo@label}{\text}{\@glo@text}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @text\endcsname\glspluralsuffix}%

```

```

        {\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @text\endcsname}%
    {\@glo@label}{first}{\@glo@first}%
\ifx\@glo@first\@gls@default@value
    \expandafter\gls@assign@field\expandafter
        {\csname glo@\@glo@label @plural\endcsname}%
        {\@glo@label}{firstpl}{\@glo@firstplural}%
\else
    \expandafter\gls@assign@field\expandafter
        {\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
        {\@glo@label}{firstpl}{\@glo@firstplural}%
\fi
\ifcsundef{@glotype@\@glo@type @counter}%
{%
    \def\@glo@defaultcounter{\glscounter}%
}%
{%
    \letcs\@glo@defaultcounter{@glotype@\@glo@type @counter}%
}%
\gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
\gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
\gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
\gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
\gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
\gls@assign@field{}{\@glo@label}{userv}{\@glo@userv}%
\gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
\gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
\gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
\gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
\gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%
\ifx\@glo@name\@glsnoname
    \@glsnoname
    \let\@glo@name\@gls@default@value
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%
\ifcsundef{glo@\@glo@label @numberlist}%
{%
    \csxdef{glo@\@glo@label @numberlist}{%
        \noexpand\@gls@missingnumberlist{\@glo@label}}%
}%
{%
    {}%
}%
\@gls@storenonumberlist{\@glo@label}%
\def\@glo@@desc{\@glo@first}%
\ifx\@glo@desc\@glo@@desc
    \let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@glsnodesc
    \@glsnodesc
    \let\@glo@desc\@gls@default@value

```

```

\fi
\gls@assign@desc{\@glo@label}%
\@gls@defsort{\@glo@type}{\@glo@label}%
\def\@glo@symbol{\@glo@text}%
\ifx\@glo@symbol\@glo@symbol
  \let\@glo@symbol\@glo@text
\fi
\gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
\expandafter
\gls@assign@field\expandafter
  {\csname glo@\@glo@label @symbol\endcsname}
  {\@glo@label}{symbolplural}{\@glo@symbolplural}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagfalse\endcsname{%
  \noexpand\global
    \noexpand\let\expandafter\noexpand
      \csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagtrue\endcsname{%
  \noexpand\global
    \noexpand\let\expandafter\noexpand
      \csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
}%
\csname glo@\@glo@label @flagfalse\endcsname
\@glo@autosee
\ifignoredglossary\@glo@type
{%
  \csdef{glo@\@glo@label @index}{}%
}
{%
  \do@glo@storeentry{\@glo@label}%
}%
\@newglossaryentry@defcounters
\@newglossaryentryposthook
}
\newcommand*{\@glo@autosee}{%
  \ifdefined\@glo@see{%
    {%
      \protected@edef\@do@glssee{%
        \noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
        \noexpand\expandafter\noexpand\@glssee\noexpand\@glo@list{\@glo@label}}%
      \@do@glssee
    }%
    \@glo@autoseehook
  }%
\newcommand*{\@glo@autoseehook}{%
\newcommand*{\@newglossaryentryprehook}{%
\newcommand*{\@newglossaryentryposthook}{%
\newcommand*{\@newglossaryentry@defcounters}{%
\newcommand*{\glsmoveentry}[2]{%
  \protected@edef\@glo@thislabel{\glsdetoklabel{#1}}%

```



```

\else
\expandafter\protected\xdef\csname glo@#1@index\endcsname{%
\csname glo@\@glo@parent @index\endcsname\@gls@levelchar
\@glo@sort\@gls@actualchar\@glo@prefix
\@glossarysubentryfield
{\csname glo@#1@level\endcsname}\@glo@esclabel}%
}%
\fi
\fi
}
\AtBeginDocument{%
\ifpackageloaded{amsmath}%
{\let\gls@ifnotmeasuring\@gls@ifnotmeasuring}%
{}}%
}
\newcommand*{\@gls@ifnotmeasuring}[1]{%
\ifmeasuring@
\else
#1%
\fi
}
\newcommand*\gls@ifnotmeasuring[1]{#1}
\def\@gls@patchtabularx#1\hbox#2#3!!{%
\def\TX@trial##1{#1\hbox{\let\glsunset\@gobble#2}#3}%
}
\newcommand*\glspatchtabularx{%
\ifdef\TX@trial
{%
\expandafter\@gls@patchtabularx\TX@trial{##1}!!%
\let\glspatchtabularx\relax
}%
{}}%
}
\newcommand*{\glsreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glsreset{#1}%
}%
}%
}
\newcommand*{\glslocalreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glslocalreset{#1}%
}%
}%
}%

```



```

}
\newcommand*{\glsunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glsunset{#1}%
    }%
  }%
}
\newcommand*{\glslocalunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glslocalunset{#1}%
    }%
  }%
}
\newcommand*{\@glslocalunset}{\@glslocalunset}
\newcommand*{\@glslocalunset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
}
\newcommand*{\@glsunset}{\@glsunset}
\newcommand*{\@glsunset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
}
\newcommand*{\@glslocalreset}{\@glslocalreset}
\newcommand*{\@glslocalreset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
}
\newcommand*{\@glsreset}{\@glsreset}
\newcommand*{\@glsreset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
}
\newcommand*{\glsresetall}[1][\@glo@types]{%
  \forallglsentries[1]{\@glsentry}%
  {%
    \glsreset{\@glsentry}%
  }%
}
\newcommand*{\glslocalresetall}[1][\@glo@types]{%
  \forallglsentries[1]{\@glsentry}%
  {%
    \glslocalreset{\@glsentry}%
  }%
}
\newcommand*{\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[1]{\@glsentry}%
  {%

```

```

        \glsunset{\@glsentry}%
    }%
}
\newcommand*{\glslocalunsetall}[1][\@glo@types]{%
    \forallglsentries[#1]{\@glsentry}%
    {%
        \glslocalunset{\@glsentry}%
    }%
}
\newcommand*{\@newglossaryentry@defcounters}{%
    \csdef{glo@\@glo@label @currcount}{0}%
    \csdef{glo@\@glo@label @prevcount}{0}%
}
\newcommand*{\glsenableentrycount}{%
    \let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
    \renewcommand*{\gls@defdocnewglossaryentry}{%
        \renewcommand*{\newglossaryentry}[2]{%
            \PackageError{glossaries}{\string\newglossaryentry\space
                may only be used in the preamble when entry counting has
                been activated}{If you use \string\glsenableentrycount\space
                you must place all entry definitions in the preamble not in
                the document environment}%
        }%
    }%
}
\newcommand*{\glsentrycurrcount}[1]{%
    \ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
    {0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*{\glsentryprevcount}[1]{%
    \ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
    {0}{\@gls@entry@field{##1}{prevcount}}%
}%
\renewcommand*{\@glsunset}[1]{%
    \@glsunset{##1}%
    \@gls@increment@currcount{##1}%
}%
\renewcommand*{\@glslocalunset}[1]{%
    \@glslocalunset{##1}%
    \@gls@local@increment@currcount{##1}%
}%
\renewcommand*{\@glsreset}[1]{%
    \@glsreset{##1}%
    \csgdef{glo@\glsdetoklabel{##1}@currcount}{0}%
}%
\renewcommand*{\@glslocalreset}[1]{%
    \@glslocalreset{##1}%
    \csdef{glo@\glsdetoklabel{##1}@currcount}{0}%
}%
\def\@cgls@##1##2[##3]{%
    \ifnum\glsentryprevcount{##2}=1\relax

```

```

\cglformat{##2}{##3}%
\glsunset{##2}%
\else
\@gls@{##1}{##2}{##3}%
\fi
}%
\def\@cglsp1@##1##2[##3]{%
\ifnum\glsentryprevcount{##2}=1\relax
\cglsp1format{##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\@cGlsp1@##1##2[##3]{%
\ifnum\glsentryprevcount{##2}=1\relax
\cGlsp1format{##2}{##3}%
\glsunset{##2}%
\else
\@Glsp1@{##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*{\cgl}{\@gls@hyp@opt\@cgl}
\newcommand*{\@cgl}[2][{}]{%
    \new@ifnextchar[{\@cgl@{\#1}{\#2}}{\@cgl@{\#1}{\#2}[]}%
}
\def\@cgl@{\#1\#2}[{\#3}{%
    \GlossariesWarning{\string\cgl\space is defaulting to
        \string\gls\space since you haven't enabled entry counting}%
    \@cgl@{\#1}{\#2}[{\#3}%
}
\newcommand*{\cglformat}[2]{%
    \ifglshaslong{\#1}{\glsentrylong{\#1}}{\glsentryfirst{\#1}}{\#2}%
}
\newrobustcmd*{\cGls}{\@gls@hyp@opt\@cGls}
\newcommand*{\@cGls}[2][{}]{%
    \new@ifnextchar[{\@cGls@{\#1}{\#2}}{\@cGls@{\#1}{\#2}[]}%
}
\def\@cGls@{\#1\#2}[{\#3}{%
    \GlossariesWarning{\string\cGls\space is defaulting to
        \string\Gls\space since you haven't enabled entry counting}%
    \@cGls@{\#1}{\#2}[{\#3}%
}
\newcommand*{\cGlsformat}[2]{%
    \ifglshaslong{\#1}{\Glsentrylong{\#1}}{\Glsentryfirst{\#1}}{\#2}%
}
\newrobustcmd*{\cglspl}{\@gls@hyp@opt\@cglspl}
\newcommand*{\@cglspl}[2][{}]{%
    \new@ifnextchar[{\@cglspl@{\#1}{\#2}}{\@cglspl@{\#1}{\#2}[]}%
}
\def\@cglspl@{\#1\#2}[{\#3}{%
    \GlossariesWarning{\string\cglspl\space is defaulting to
        \string\glspl\space since you haven't enabled entry counting}%
    \@cglspl@{\#1}{\#2}[{\#3}%
}
\newcommand*{\cglsplformat}[2]{%
    \ifglshaslong{\#1}{\glsentrylongpl{\#1}}{\glsentryfirstplural{\#1}}{\#2}%
}
\newrobustcmd*{\cGlspl}{\@gls@hyp@opt\@cGlspl}
\newcommand*{\@cGlspl}[2][{}]{%
    \new@ifnextchar[{\@cGlspl@{\#1}{\#2}}{\@cGlspl@{\#1}{\#2}[]}%
}
\def\@cGlspl@{\#1\#2}[{\#3}{%
    \GlossariesWarning{\string\cGlspl\space is defaulting to
        \string\Glspl\space since you haven't enabled entry counting}%
    \@cGlspl@{\#1}{\#2}[{\#3}%
}

```

```

}
\newcommand*{\cGlsplformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
}
\newcommand*{\loadglsentries}[2][\@gls@default]{%
  \let\@gls@default\glsdefaulttype
  \def\glsdefaulttype{#1}\input{#2}%
  \let\glsdefaulttype\@gls@default
}
\@onlypreamble{\loadglsentries}
\newcommand*{\glstextformat}[1]{#1}
\newcommand*{\glsentryfmt}{%
  \@gls@default@entryfmt\glsdisplayfirst\glsdisplay
}
\newcommand*{\@gls@default@entryfmt}[2]{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          #2{\glsentryplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
        }%
        {%
          #1{\glsentryfirstplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
        }%
      }%
    }%
    {%
      \ifglsused\glslabel
      {%
        \ifbool{glscompatible-3.07}%
        {%
          \protected@edef\@glo@etext{%
            #2{\glsentryplural{\glslabel}}%
              {\glsentrydescplural{\glslabel}}%
              {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
          \xmakefirstuc\@glo@etext
        }%
        {%
          #2{\Glsentryplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
          }%
        }%
      }%
    }%
  }%

```

```

{%
  \ifbool{glscompatible-3.07}%
  {%
    \protected@edef\@glo@etext{%
      #1{\glsentryfirstplural{\glslabel}}%
      {\glsentrydescplural{\glslabel}}%
      {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
    \xmakefirstuc\@glo@etext
  }%
  {%
    #1{\Glsentryfirstplural{\glslabel}}%
    {\glsentrydescplural{\glslabel}}%
    {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
  }%
}%
}%
{%
  \ifglsused\glslabel
  {%
    \mfirstucMakeUppercase{#2{\glsentryplural{\glslabel}}%
      {\glsentrydescplural{\glslabel}}%
      {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
  }%
  {%
    \mfirstucMakeUppercase{#1{\glsentryfirstplural{\glslabel}}%
      {\glsentrydescplural{\glslabel}}%
      {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
  }%
}%
}%
{%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      #2{\glsentrytext{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
    {%
      #1{\glsentryfirst{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
  }%
}%
{%
  \ifglsused\glslabel
  {%
    \ifbool{glscompatible-3.07}%
    {%

```

```

\protected@edef\@glo@etext{%
  #2{\glsentrytext{\glslabel}}%
  {\glsentrydesc{\glslabel}}%
  {\glsentrysymbol{\glslabel}}{\glsinsert}}%
\makefirstuc\@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}}%
    \makefirstuc\@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}}{\glsinsert}%
}%

```

```

    {%
      #1{\glscustomtext}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{%}%
    }%
  }%
}
\newcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glusifplural
    {%
      \glscapscase
      {%
        \ifglssused\glslabel
        {%
          \glsentryplural{\glslabel}\glsinsert
        }%
        {%
          \glsentryfirstplural{\glslabel}\glsinsert
        }%
      }%
    }%
    {%
      \ifglssused\glslabel
      {%
        \Glsentryplural{\glslabel}\glsinsert
      }%
      {%
        \Glsentryfirstplural{\glslabel}\glsinsert
      }%
    }%
    {%
      \ifglssused\glslabel
      {%
        \mfirstucMakeUppercase
        {\glsentryplural{\glslabel}\glsinsert}%
      }%
      {%
        \mfirstucMakeUppercase
        {\glsentryfirstplural{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glscapscase
    {%
      \ifglssused\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
}

```

```

    }%
  }%
}%
\@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\theHglentrycounter{@gls@Hcounter}%
\restorecounters@
\fi
}%
}%
{%
\edef\theglsentrycounter{\expandafter\noexpand
\csname the@gls@counter\endcsname}%
}%
\ifx{@gls@Hcounter}\@empty
\ifcsundef{theH@gls@counter}%
{%
\def\theHglentrycounter{\theglsentrycounter}%
}%
{%
\protected@edef\theHglentrycounter{\expandafter\noexpand
\csname theH@gls@counter\endcsname}%
}%
}

```

```

\fi
}
\def\@set@glo@numformat#1#2#3#4{%
\expandafter\@glo@check@mkidxrangechar#3\@nil
\protected@edef#1{%
\@glo@prefix setentrycounter[#4]{#2}%
\expandafter\string\csname\@glo@suffix\endcsname
}%
\@gls@checkmkidxchars#1%
}
\def\@glo@check@mkidxrangechar#1#2\@nil{%
\if#1(\relax
\def\@glo@prefix{)%
\if\relax#2\relax
\def\@glo@suffix{glsnumberformat}%
\else
\def\@glo@suffix{#2}%
\fi
\else
\if#1)\relax
\def\@glo@prefix{)}%
\if\relax#2\relax
\def\@glo@suffix{glsnumberformat}%
\else
\def\@glo@suffix{#2}%
\fi
\else
\def\@glo@prefix{}\def\@glo@suffix{#1#2}%
\fi
\fi}
\newcommand*{\@gls@escbsdq}[1]{%
\def\@gls@checkedmkidx{}%
\let\gls@xdystring=#1\relax
\@onelevel@sanitize\gls@xdystring
\edef\do@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null}%
\do@gls@xdycheckbackslash
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\@for\@gls@tmp:=\gls@protected@pagefmts\do
{%
\edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\expandonce\@gls@tmp}%
\@onelevel@sanitize\@gls@sanitized@tmp
\edef\gls@dostubst{%
\noexpand\DTLsubstituteall\noexpand\gls@xdystring
{\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
}%
}

```

```

\gls@dostsubst
}%
\let#1=\gls@xdyststring
}
\newcommand{\@gls@checkmkidxchars}[1]{%
\ifglsxindy
\@gls@escbsdq{#1}%
\else
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkquote#1\@nil""\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkescquote#1\@nil""\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkescactual#1\@nil"??"\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkactual#1\@nil??\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkbar#1\@nil||\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkescbar#1\@nil|||\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checklevel#1\@nil!!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkesclevel#1\@nil\!\!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\edef\@do@gls@checkesc{%
\noexpand\@gls@checkesc\expandonce{#1}\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\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
    \fi
}

```

```

\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\"@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!\!\null}%
\else
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!#3\null}%
\fi
\fi
\fi
\@gls@checkesclevel
}
\def\@gls@checkbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@gls@checkbar{\@gls@checkbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkbar{\@gls@checkbar#2||\null}%
\else
\def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}
\def\@gls@checklevel#1!#2!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checklevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
\def\@gls@checklevel{\@gls@checklevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar}%

```

```

\ifx\null#3\null
  \def\@gls@checklevel{\@gls@checklevel#2!\null}%
\else
  \def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
\fi
\fi
\@gls@checklevel
}
\def\@gls@checkactual#1?#2?#3\null{%
  \gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkactual{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
      \def\@gls@checkactual{\@gls@checkactual#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@gls@actualchar}%
    \ifx\null#3\null
      \def\@gls@checkactual{\@gls@checkactual#2??\null}%
    \else
      \def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
    \fi
  \fi
  \@gls@checkactual
}
\bgroup
\catcode'\0\relax
\catcode'\12\relax
\gdef\@gls@checkesc#1\#2\#3\null{%
  \gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkesc{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@backslashchar\@gls@quotechar\@backslashchar}%
      \def\@gls@checkesc{\@gls@checkesc#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@backslashchar}%
    \ifx\null#3\null

```

```

        |def|@@gls@checkesc{|@gls@checkesc#2\\|null}%
    |else
        |def|@@gls@checkesc{|@gls@checkesc#2\#3|null}%
    |fi
|fi
|@@gls@checkesc
}
|egroup
\def\@gls@xdycheckquote#1"#2"#3\null{%
    \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
    \toks@={#1}%
    \ifx\null#2\null
        \ifx\null#3\null
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
            \def\@gls@xdycheckquote{\relax}%
        |else
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
                \string"\string"}%
            \def\@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
        \fi
    |else
        \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
            \string"}%
        \ifx\null#3\null
            \def\@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
        |else
            \def\@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
        \fi
    \fi
    \@gls@xdycheckquote
}
\edef\def\@gls@xdycheckbackslash{%
    \noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
        ##2\@backslashchar##3\noexpand\null{%
        \noexpand\@gls@tmpb=\noexpand\expandafter
            {\noexpand\@gls@checkedmkidx}%
        \noexpand\toks@={##1}%
        \noexpand\ifx\noexpand\null##2\noexpand\null
            \noexpand\ifx\noexpand\null##3\noexpand\null
                \noexpand\edef\noexpand\@gls@checkedmkidx{%
                    \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
                \noexpand\def\noexpand\@gls@xdycheckbackslash{\relax}%
            \noexpand|else
                \noexpand\edef\noexpand\@gls@checkedmkidx{%
                    \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
                    \@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
            \noexpand\def\noexpand\@gls@xdycheckbackslash{%
                \noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
            \noexpand\fi
        \noexpand|else

```

```

\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\glS@ifplural\@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\glS@ifplural\@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@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
\let\gl@sifplural\@firstoftwo
\let\gl@scapscase\@secondofthree
\let\gl@customtext\@empty
\def\gl@insert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gl@link[#1]{#2}{\@glo@text}%
\ifKV@gl@link@local
  \gl@localunset{#2}%
\else
  \gl@unset{#2}%
\fi
}%
\gl@postlinkhook
}
\newrobustcmd*{\GLSp1}{\@gl@hyp@opt\@GLSp1}
\newcommand*{\@GLSp1}[2][{}]{%
  \new@ifnextchar[{\@GLSp1@{#1}{#2}}{\@GLSp1@{#1}{#2}[]}%
}
\def\@GLSp1@#1#2[#3]{%
  \gl@doifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\gl@sifplural\@firstoftwo
    \let\gl@scapscase\@thirdofthree
    \let\gl@customtext\@empty
    \def\gl@insert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \ifKV@gl@link@local
      \gl@localunset{#2}%
    \else
      \gl@unset{#2}%
    \fi
  }%
  \gl@postlinkhook
}
\newrobustcmd*{\gl@disp}{\@gl@hyp@opt\@gl@disp}
\newcommand*{\@gl@disp}[3][{}]{%
  \gl@doifexists{#2}{%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\gl@sifplural\@secondoftwo
    \let\gl@scapscase\@firstofthree
    \def\gl@customtext{#3}%
    \def\gl@insert{}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \ifKV@gl@link@local
      \gl@localunset{#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}{\glstentrytext{#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{\glstentrytext{#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}{\Glstentrytext{#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}{\glstentryfirst{#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{\glstentryfirst{#2}{#3}}%
}
}

```

```

\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
\newcommand*{\@glsplural}[2] [] {%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} [] }}
\def\@glsplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\newcommand*{\@Glsplural}[2] [] {%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} [] }}
\def\@Glsplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
\newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
\newcommand*{\@GLSplural}[2] [] {%
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} [] }}
\def\@GLSplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{\@glsfirstplural}[2] [] {%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} [] }}
\def\@glsfirstplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
\newcommand*{\@Glsfirstplural}[2] [] {%
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2} [] }}
\def\@Glsfirstplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
\newcommand*{\@GLSfirstplural}[2] [] {%
  \new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} [] }}
\def\@GLSfirstplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
\newcommand*{\@glsname}[2] [] {%
  \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2} [] }}
\def\@glsname@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryname{#2}#3}%
}
\newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}
\newcommand*{\@Glsname}[2] [] {%
  \new@ifnextchar[{\@Glsname@{#1}{#2}}{\@Glsname@{#1}{#2} [] }}
\def\@Glsname@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryname{#2}#3}%
}
\newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}
\newcommand*{\@GLSname}[2] [] {%

```

```

\new@ifnextchar [{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}}
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryname{#2}#3}}%
}
\newrobustcmd*{\glsdesc}{\@gls@hyp@opt\@glsdesc}
\newcommand*{\@glsdesc}[2] [] {%
  \new@ifnextchar [{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2} []}}
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydesc{#2}#3}%
}
\newrobustcmd*{\Glsdesc}{\@gls@hyp@opt\@Glsdesc}
\newcommand*{\@Glsdesc}[2] [] {%
  \new@ifnextchar [{\@Glsdesc@{#1}{#2}}{\@Glsdesc@{#1}{#2} []}}
\def\@Glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%
}
\newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\newcommand*{\@GLSdesc}[2] [] {%
  \new@ifnextchar [{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}}
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydesc{#2}#3}}%
}
\newrobustcmd*{\glsdescplural}{\@gls@hyp@opt\@glsdescplural}
\newcommand*{\@glsdescplural}[2] [] {%
  \new@ifnextchar [{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}}
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\Glsdescplural}{\@gls@hyp@opt\@Glsdescplural}
\newcommand*{\@Glsdescplural}[2] [] {%
  \new@ifnextchar [{\@Glsdescplural@{#1}{#2}}{\@Glsdescplural@{#1}{#2} []}}
\def\@Glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\newcommand*{\@GLSdescplural}[2] [] {%
  \new@ifnextchar [{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}}
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydescplural{#2}#3}}%
}
\newrobustcmd*{\glssymbol}{\@gls@hyp@opt\@glssymbol}
\newcommand*{\@glssymbol}[2] [] {%
  \new@ifnextchar [{\@glssymbol@{#1}{#2}}{\@glssymbol@{#1}{#2} []}}
\def\@glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\Glsymbol}{\@gls@hyp@opt\@Glsymbol}
\newcommand*{\@Glsymbol}[2] [] {%
  \new@ifnextchar [{\@Glsymbol@{#1}{#2}}{\@Glsymbol@{#1}{#2} []}}
\def\@Glsymbol@#1#2[#3]{%

```

```

\@gls@field@link{#1}{#2}{\Glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\GLSsymbol}{\@gls@hyp@opt\@GLSsymbol}
\newcommand*{\@GLSsymbol}[2][]{%
  \new@ifnextchar[{\@GLSsymbol@{#1}{#2}}{\@GLSsymbol@{#1}{#2}[]}}
\def\@GLSsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrysymbol{#2}#3}}%
}
\newrobustcmd*{\glssymbolplural}{\@gls@hyp@opt\@glssymbolplural}
\newcommand*{\@glssymbolplural}[2][]{%
  \new@ifnextchar[{\@glssymbolplural@{#1}{#2}}{\@glssymbolplural@{#1}{#2}[]}}
\def\@glssymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\Glsymbolplural}{\@gls@hyp@opt\@Glsymbolplural}
\newcommand*{\@Glsymbolplural}[2][]{%
  \new@ifnextchar[{\@Glsymbolplural@{#1}{#2}}{\@Glsymbolplural@{#1}{#2}[]}}
\def\@Glsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
\newcommand*{\@GLSsymbolplural}[2][]{%
  \new@ifnextchar[{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2}[]}}
\def\@GLSsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrysymbolplural{#2}#3}}%
}
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
\newcommand*{\@glsuseri}[2][]{%
  \new@ifnextchar[{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2}[]}}
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
\newcommand*{\@Glsuseri}[2][]{%
  \new@ifnextchar[{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2}[]}}
\def\@Glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\newcommand*{\@GLSuseri}[2][]{%
  \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2}[]}}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseri{#2}#3}}%
}
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
\newcommand*{\@glsuserii}[2][]{%
  \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2}[]}}
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}

```

```

\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
\newcommand*{\@Glsuserii}[2][]{\%
  \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2}[]}}
\def\@Glsuserii@#1#2[#3]{\%
  \@gls@field@link{#1}{#2}{\@Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
\newcommand*{\@Glsuserii}[2][]{\%
  \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2}[]}}
\def\@Glsuserii@#1#2[#3]{\%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\@Glsentryuserii{#2}#3}}%
}
\newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
\newcommand*{\@Glsuseriii}[2][]{\%
  \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2}[]}}
\def\@Glsuseriii@#1#2[#3]{\%
  \@gls@field@link{#1}{#2}{\@Glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
\newcommand*{\@Glsuseriii}[2][]{\%
  \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2}[]}}
\def\@Glsuseriii@#1#2[#3]{\%
  \@gls@field@link{#1}{#2}{\@Glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
\newcommand*{\@Glsuseriii}[2][]{\%
  \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2}[]}}
\def\@Glsuseriii@#1#2[#3]{\%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\@Glsentryuseriii{#2}#3}}%
}
\newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
\newcommand*{\@Glsuseriv}[2][]{\%
  \new@ifnextchar[{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2}[]}}
\def\@Glsuseriv@#1#2[#3]{\%
  \@gls@field@link{#1}{#2}{\@Glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
\newcommand*{\@Glsuseriv}[2][]{\%
  \new@ifnextchar[{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2}[]}}
\def\@Glsuseriv@#1#2[#3]{\%
  \@gls@field@link{#1}{#2}{\@Glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
\newcommand*{\@Glsuseriv}[2][]{\%
  \new@ifnextchar[{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2}[]}}
\def\@Glsuseriv@#1#2[#3]{\%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\@Glsentryuseriv{#2}#3}}%
}
\newrobustcmd*{\Glsuserv}{\@gls@hyp@opt\@Glsuserv}
\newcommand*{\@Glsuserv}[2][]{\%

```

```

\new@ifnextchar [{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2} []}}
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\Glsuserv}{\@gls@hyp@opt\@Glsuserv}
\newcommand*{\@Glsuserv}[2] [] {%
  \new@ifnextchar [{\@Glsuserv@{#1}{#2}}{\@Glsuserv@{#1}{#2} []}}
\def\@Glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
\newcommand*{\@GLSuserv}[2] [] {%
  \new@ifnextchar [{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2} []}}
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuserv{#2}#3}}%
}
\newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
\newcommand*{\@glsuservi}[2] [] {%
  \new@ifnextchar [{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2} []}}
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
\newrobustcmd*{\Glsuservi}{\@gls@hyp@opt\@Glsuservi}
\newcommand*{\@Glsuservi}[2] [] {%
  \new@ifnextchar [{\@Glsuservi@{#1}{#2}}{\@Glsuservi@{#1}{#2} []}}
\def\@Glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
\newrobustcmd*{\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\glsapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\Glsentryshort{#2}}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrshortpl}{\@gls@hyp@opt\@ns@acrshortpl}
\newcommand*{\ns@acrshortpl}[2] [] {%
  \new@ifnextchar[{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} []}]%
}
\def\@acrshortpl#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo

```



```

\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \acronymfont{\glentryshortpl{#2}}#3%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\@ns@Acrshortpl}
\newcommand*{\ns@Acrshortpl}[2] [] {%
  \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2} []}%
}
\def\@Acrshortpl#1#2[#3] {%
  \glstoifexists{#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] {%
  \glstoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\glentryshortpl{#2}}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrlong}{\@gls@hyp@opt\@ns@acrlong}
\newcommand*{\ns@acrlong}[2] [] {%
  \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2} []}%
}

```

```

}
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@glsl@link@checkfirsthyper\@glsl@link@nocheckfirsthyper
    \def\glsl@label{#2}%
    \let\glsl@ifplural\@secondoftwo
    \let\glsl@capscase\@firstofthree
    \let\glsl@insert\@empty
    \def\glsl@customtext{%
      \glsl@entrylong{#2}#3%
    }%
    \@glsl@link{#1}{#2}{\csname glsl@glstype @entryfmt\endcsname}%
  }%
  \glsl@postlinkhook
}
\newrobustcmd*{\Acrlong}{\@glsl@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@glsl@link@checkfirsthyper\@glsl@link@nocheckfirsthyper
    \def\glsl@label{#2}%
    \let\glsl@ifplural\@secondoftwo
    \let\glsl@capscase\@secondofthree
    \let\glsl@insert\@empty
    \def\glsl@customtext{%
      \glsl@entrylong{#2}#3%
    }%
    \@glsl@link{#1}{#2}{\csname glsl@glstype @entryfmt\endcsname}%
  }%
  \glsl@postlinkhook
}
\newrobustcmd*{\ACrlong}{\@glsl@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@glsl@link@checkfirsthyper\@glsl@link@nocheckfirsthyper
    \def\glsl@label{#2}%
    \let\glsl@ifplural\@secondoftwo
    \let\glsl@capscase\@thirdofthree
    \let\glsl@insert\@empty
    \def\glsl@customtext{%
      \mfirstucMakeUppercase{\glsl@entrylong{#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@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@name}%
    \expandafter\@gls@getbody\@glo@text{}\@nil
    \expandafter\ifx\@gls@body\glsentrylong\relax
    \expandafter\Glsentrylong\@gls@rest
  }%
  \else

```

```

\expandafter\ifx\@gls@body\glsentryshort\relax
\expandafter\Glsentryshort\@gls@rest
\else
\expandafter\ifx\@gls@body\acronymfont\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\xmakefirstuc{\@glo@text}%
\fi
\fi
\fi
}%
{%
\@Gls@entry@field{#1}{name}%
}%
}
\newcommand*{\glsentrydesc}[1]{\@Gls@entry@field{#1}{desc}}
\newrobustcmd*{\Glsentrydesc}[1]{%
\@Gls@entry@field{#1}{desc}%
}
\newcommand*{\glsentrydescplural}[1]{%
\@Gls@entry@field{#1}{descplural}%
}
\newrobustcmd*{\Glsentrydescplural}[1]{%
\@Gls@entry@field{#1}{descplural}%
}
\newcommand*{\glsentrytext}[1]{\@Gls@entry@field{#1}{text}}
\newrobustcmd*{\Glsentrytext}[1]{%
\@Gls@entry@field{#1}{text}%
}
\newcommand*{\glsentryplural}[1]{%
\@Gls@entry@field{#1}{plural}%
}
\newrobustcmd*{\Glsentryplural}[1]{%
\@Gls@entry@field{#1}{plural}%
}
\newcommand*{\glsentrysymbol}[1]{%
\@Gls@entry@field{#1}{symbol}%
}
\newrobustcmd*{\Glsentrysymbol}[1]{%
\@Gls@entry@field{#1}{symbol}%
}

```

```

}
\newcommand*{\glsentrysymbolplural}[1]{%
  \@gls@entry@field{#1}{symbolplural}%
}
\newrobustcmd*{\Glsentrysymbolplural}[1]{%
  \@Gls@entry@field{#1}{symbolplural}%
}
\newcommand*{\glsentryfirst}[1]{%
  \@gls@entry@field{#1}{first}%
}
\newrobustcmd*{\Glsentryfirst}[1]{%
  \@Gls@entry@field{#1}{first}%
}
\newcommand*{\glsentryfirstplural}[1]{%
  \@gls@entry@field{#1}{firstpl}%
}
\newrobustcmd*{\Glsentryfirstplural}[1]{%
  \@Gls@entry@field{#1}{firstpl}%
}
\newcommand{\glscapitalisewords}[1]{%
  \capitalisewords{#1}%
}
\newrobustcmd*{\@glsentrytitlecase}[2]{%
  \glsdoifexists{#1}%
  {%
    \glsfieldfetch{#1}{#2}{\@gls@value}%
    \expandafter\glscapitalisewords\expandafter{\@gls@value}%
  }%
}
\ifdef\texorpdfstring
{
  \newcommand*{\glsentrytitlecase}[2]{%
    \texorpdfstring
      {\@glsentrytitlecase{#1}{#2}}%
      {\@gls@entry@field{#1}{#2}}%
  }
}
{
  \newcommand*{\glsentrytitlecase}[2]{\@glsentrytitlecase{#1}{#2}}
}
\newcommand*{\glsentrytype}[1]{\@gls@entry@field{#1}{type}}
\newcommand*{\glsentrysort}[1]{%
  \@gls@entry@field{#1}{sort}%
}
\newcommand*{\glsentryparent}[1]{%
  \@gls@entry@field{#1}{parent}%
}
\newcommand*{\glsentryuseri}[1]{%
  \@gls@entry@field{#1}{useri}%
}

```

```

\newrobustcmd*{\Glsentryuseri}[1]{%
  \@Gls@entry@field{#1}{useri}%
}
\newcommand*{\glsentryuserii}[1]{%
  \@gls@entry@field{#1}{userii}%
}
\newrobustcmd*{\Glsentryuserii}[1]{%
  \@Gls@entry@field{#1}{userii}%
}
\newcommand*{\glsentryuseriii}[1]{%
  \@gls@entry@field{#1}{useriii}%
}
\newrobustcmd*{\Glsentryuseriii}[1]{%
  \@Gls@entry@field{#1}{useriii}%
}
\newcommand*{\glsentryuseriv}[1]{%
  \@gls@entry@field{#1}{useriv}%
}
\newrobustcmd*{\Glsentryuseriv}[1]{%
  \@Gls@entry@field{#1}{useriv}%
}
\newcommand*{\glsentryuserv}[1]{%
  \@gls@entry@field{#1}{userv}%
}
\newrobustcmd*{\Glsentryuserv}[1]{%
  \@Gls@entry@field{#1}{userv}%
}
\newcommand*{\glsentryuservi}[1]{%
  \@gls@entry@field{#1}{uservi}%
}
\newrobustcmd*{\Glsentryuservi}[1]{%
  \@Gls@entry@field{#1}{uservi}%
}
\newcommand*{\glsentryshort}[1]{\@gls@entry@field{#1}{short}}
\newrobustcmd*{\Glsentryshort}[1]{%
  \@Gls@entry@field{#1}{short}%
}
\newcommand*{\glsentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}
\newrobustcmd*{\Glsentryshortpl}[1]{%
  \@Gls@entry@field{#1}{shortpl}%
}
\newcommand*{\glsentrylong}[1]{\@gls@entry@field{#1}{long}}
\newrobustcmd*{\Glsentrylong}[1]{%
  \@Gls@entry@field{#1}{long}%
}
\newcommand*{\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}
\newrobustcmd*{\Glsentrylongpl}[1]{%
  \@Gls@entry@field{#1}{longpl}%
}
\newcommand*{\glsentryfull}[1]{%

```

```

\acrfullformat{\glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
}
\newrobustcmd*{\Glsentryfull}[1]{%
\acrfullformat{\Glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
}
\newcommand*{\glsentryfullpl}[1]{%
\acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newrobustcmd*{\Glsentryfullpl}[1]{%
\acrfullformat{\Glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newcommand*{\glsentrynumberlist}[1]{%
\glsdoifexists{#1}%
{%
\@gls@entry@field{#1}{numberlist}%
}%
}
\@ifpackageloaded{hyperref}{%
\newcommand*{\glsdisplaynumberlist}[1]{%
\GlossariesWarning
{%
\string\glsdisplaynumberlist\space
doesn't work with hyperref.^^JUsing
\string\glsentrynumberlist\space instead%
}%
\glsentrynumberlist{#1}%
}%
}%
{%
\newcommand*{\glsdisplaynumberlist}[1]{%
\glsdoifexists{#1}%
{%
\bgroup
\protected@edef\@glo@label{\glsdetoklabel{#1}}%
\let\@org@glsnumberformat\glsnumberformat
\def\glsnumberformat##1{##1}%
\protected@edef\the@numberlist{%
\csname glo@\@glo@label @numberlist\endcsname}%
\def\@gls@numlist@sep{}%
\def\@gls@numlist@nextsep{}%
\def\@gls@numlist@lastsep{}%
\def\@gls@thislist{}%
\def\@gls@donext@def{}%
\renewcommand\do[1]{%
\protected@edef\@gls@thislist{%
\@gls@thislist
\noexpand\@gls@numlist@sep
##1%
}%
\let\@gls@numlist@sep\@gls@numlist@nextsep

```



```

\def\@gls@numlist@nextsep{\glsnumlistsep}%
\@gls@donext@def
\def\@gls@donext@def{%
\def\@gls@numlist@lastsep{\glsnumlistlastsep}%
}%
}%
\expandafter \glsnumlistparser \expandafter{\the@numberlist}%
\let\@gls@numlist@sep\@gls@numlist@lastsep
\@gls@thislist
\egroup
}%
}
}
\newcommand*{\glsnumlistsep}{, }
\newcommand*{\glsnumlistlastsep}{ \& }
\newcommand*{\gls hyperlink}[2][\glsentrytext{\@glo@label}]{%
\def\@glo@label{#2}%
\@glslink{\glo linkprefix\glsdetoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*{\glsadd}[2][]{%
\@gls@adjustmode
\glsdoifexists{#2}%
{%
\def\@glsnumberformat{glsnumberformat}%
\protected@edef\@gls@counter{\csname glo@%glsdetoklabel{#2}@counter\endcsname}%
\setkeys{glossadd}{#1}%
\@gls@saveentrycounter
\@gls@setsort{#2}%
\@do@wrglossary{#2}%
}%
}
\newcommand*{\@gls@adjustmode}{}
\AtBeginDocument{\renewcommand*{\@gls@adjustmode}{\ifvmode\mbox{}\fi}}
\newrobustcmd*{\glsaddall}[1][]{%
\protected@edef\@glo@type{\@glo@types}%
\setkeys{glossadd}{#1}%
\forallglsentries[\@glo@type]{\@glo@entry}{%
\glsadd[#1]{\@glo@entry}%
}%
}
\newrobustcmd*{\glsaddallunused}[1][\@glo@types]{%
\forallglsentries[#1]{\@glo@entry}%
{%
\ifglsused{\@glo@entry}{\@glsadd[format=glsignore]{\@glo@entry}}%
}%
}
\newcommand*{\glsignore}[1]{}
\edef\glsopenbrace{\expandafter\@gobble\string\{}}

```

```

\edef\glsclosebrace{\expandafter\@gobble\string\}}
\edef\glsbackslash{\expandafter\@gobble\string\}
\edef\glquote#1{\string"#1\string"}
\edef\glpercentchar{\expandafter\@gobble\string\%}
\edef\glstildechar{\string~}
\ifglxindy
  \newcommand*{\@glsfirstletter}{A}
\fi
\newcommand*{\GlsSetXdyFirstLetterAfterDigits}{%
  \@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
\ifglxindy
  \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*{\@glsfirstletter}{#1}}
  \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*{\@glsfirstletter}{#1}%
    \@onelevel@sanitize\@glsfirstletter
  }
\else
  \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \glsnxindywarning\GlsSetXdyFirstLetterAfterDigits}
  \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%
    \@GlsSetXdyFirstLetterAfterDigits
  }
\fi
\ifglxindy
  \newcommand*{\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
\fi
\newcommand*{\GlsSetXdyNumberGroupOrder}[1]{%
  \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder}
\ifglxindy
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
  }
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
    \@onelevel@sanitize\@xdynumbergrouporder
  }
\else
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \glsnxindywarning\GlsSetXdyNumberGroupOrder}
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
    \@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*{\@glsminrange}{2}
\ifglxindy
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \renewcommand*{\@glsminrange}{#1}}
\else
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%

```

```

\glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglxindy
\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 \@glxminrange^^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\glssseeformat\string"
:close \string"{}\string")}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
(\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
:open \string"\string
\glossarysection[\string\glossarytoctitle]{\string
\glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
{%
\@for\@this@attr:=\@xdyattributelist\do{%
\protected\write\glswrite{}\{\string\providecommand*%
\expandafter\string
\csname glsX\@this@ctr X\@this@attr\endcsname[2]%
{%
\string\setentrycounter
[\expandafter\@gobble\string\#1]{\@this@ctr}%
\expandafter\string
\csname\@this@attr\endcsname
{\expandafter\@gobble\string\#2}%
}%
}%
}%
}%
\write\glswrite{%
\string\begin
{theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
\space\space:close \string"\glpercentchar\glstildechar n\string
\end{theglossary}\string\glossarypostamble
\glstildechar n\string" ^^J\space\space\space
:tree)}%
\write\glswrite{(markup-letter-group-list

```

```

        :sep \string"\string\glsgroupskip\glstildechar n\string"}}%
\write\glswrite{(markup-indexentry
        :open \string"\string\relax \string\glsresetentrylist
        \glstildechar n\string"}}%
\write\glswrite{(markup-locclass-list :open
        \string"\glsopenbrace\string\glossaryentrynumbers
        \glsopenbrace\string\relax\space \string"^^J\space\space\space
        :sep \string", \string"
        :close \string"\glsclosebrace\glsclosebrace\string"}}%
\write\glswrite{(markup-locref-list
        :sep \string"\string\delimN\space\string"}}%
\write\glswrite{(markup-range
        :sep \string"\string\delimR\space\string"}}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
        \write\glswrite{(markup-range
                :close "\gls@suffixF" :length 1 :ignore-end)}}%
\fi
\ifx\gls@suffixFF\@empty
\else
        \write\glswrite{(markup-range
                :close "\gls@suffixFF" :length 2 :ignore-end)}}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
        :sep \string"\string\glsgroupskip\glstildechar n\string"}}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
        :open-head \string"\string\glsgroupheading
        \glsopenbrace\string"^^J\space\space\space
        :close-head \string"\glsclosebrace\string"}}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}%
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax

```

```

\ifundef{\glswrite}{\newwrite\glswrite}{\relax
\openout\glswrite=\istfilename
\write\glswrite{\glpercentchar\space makeindex style file
  created by the glossaries package}
\write\glswrite{\glpercentchar\space for document
  '\jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\@gls@actualchar'}
\write\glswrite{encap '\@gls@encapchar'}
\write\glswrite{level '\@gls@levelchar'}
\write\glswrite{quote '\@gls@quotechar'}
\write\glswrite{keyword \string\string\glossaryentry\string}
\write\glswrite{preamble \string\string\glossarysection[\string
  \glossarytoctitle]{\string\glossarytitle}\string
  \glossarypreamble\string\n\string\begin{theglossary}\string
  \glossaryheader\string\n\string}
\write\glswrite{postamble \string\string%\string\n\string
  \end{theglossary}\string\glossarypostamble\string\n
  \string}
\write\glswrite{group_skip \string\string\glsgroupskip\string\n
  \string}
\write\glswrite{item_0 \string\string%\string\n\string}
\write\glswrite{item_1 \string\string%\string\n\string}
\write\glswrite{item_2 \string\string%\string\n\string}
\write\glswrite{item_01 \string\string%\string\n\string}
\write\glswrite{item_x1
  \string\string\relax \string\glsresetentrylist\string\n
  \string}
\write\glswrite{item_12 \string\string%\string\n\string}
\write\glswrite{item_x2
  \string\string\relax \string\glsresetentrylist\string\n
  \string}
\write\glswrite{delim_0 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_1 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_2 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_t \string\string\}\string\}\string}
\write\glswrite{delim_n \string\string\delimN \string}
\write\glswrite{delim_r \string\string\delimR \string}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string\string\glsgroupheading\string\{\string}
\write\glswrite{heading_suffix
  \string\string\}\string\relax
  \string\glsresetentrylist \string}
\write\glswrite{symhead_positive \string\glsymbols\string}
\write\glswrite{numhead_positive \string\glsnumbers\string}
\write\glswrite{page_compositor \string\glscompositor\string}
\@gls@escbsdq\gls@suffixF

```

```

\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\@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@quotearch\noexpand\@gls@quotearch}%
\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\@gls@openfile\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*{\@gls@openfile}[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@istfilename\space
-C \@gls@codepage\space
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}}%
}%
\@gls@dothiswrite
}%
{%
\GlossariesWarning{Can't 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
\ifglxindy
\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{\@gls@order}{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{%
  \ifempty{\@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
  \iflssavenumberlist
    \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}{%
  \foralllglossaries{\@glo@type}{%
    \ifcsundef{glo@\@glo@type @filetok}%
    {%
      \def\gls@tmp{}%
    }%
    {%
      \edef\gls@tmp{\expandafter\the
        \csname glo@\@glo@type @filetok\endcsname}%
    }%
    \ifx\gls@tmp\@empty
      \ifx\@glo@type\glsdefaulttype
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries.^^JRemember to use package option 'nomain' if
you
          don't want to^^Juse the main glossary}%
      \else
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries}%
      \fi
    \else
      \@glsopenfile{\glswrite}{\@glo@type}%
      \immediate\write\glswrite{%

```

```

        \expandafter\the
        \csname glo@%glo@type @filetok\endcsname}%
    \immediate\closeout\glswrite
\fi
}%
}
\newcommand*{\gls@glossary}[1]{%
    \@gls@glossary{#1}%
}
\newcommand*{\@gls@glossary}[2]{%
    \if@gls@debug
        \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
    \fi
}
\newcommand{\@gls@renewglossary}{%
    \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
    \let\@gls@renewglossary\@empty
}
\newcommand*{\gls@wrglossary}[2]{%
    \ifglssavewrites
        \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
        \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
        \expandafter{\@gls@tmp^^J}%
    \else
        \ifcsdef{glo@#1@file}%
            {%
                \expandafter\protected@write\csname glo@#1@file\endcsname{%
                    \gls@disablepagerefexpansion}{#2}%
                }%
            {%
                \ifignoredglossary{#1}{}%
                    {%
                        \GlossariesWarning{No file defined for glossary ‘#1’}%
                    }%
                }%
            }%
        \fi
    \endgroup\@esphack
}
\newcommand*{\@do@wrglossary}[1]{%
    \glswriteentry{#1}{\@do@wrglossary{#1}}%
}
\newcommand*{\gls@writeentry}[2]{%
    \ifglsindexonlyfirst
        \ifglsused{#1}{#2}%
    \else
        #2%
    \fi
}
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
    \gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}

```

```

\newcommand*{\gls@disablepagerefexpansion}{%
  \@for\@gls@this:=\gls@protected@pagefmts\do
  {%
    \expandafter\let\@gls@this\relax
  }%
}
\newcommand*{\gls@alphpage}{\@alph\c@page}
\newcommand*{\gls@Alphpage}{\@Alph\c@page}
\newcommand*{\gls@numberpage}{\number\c@page}
\newcommand*{\gls@arabicpage}{\@arabic\c@page}
\newcommand*{\gls@romanpage}{\romannumeral\c@page}
\newcommand*{\gls@Romanpage}{\@Roman\c@page}
\newcommand*{\gls@saddprotectedpagefmt}[1]{%
  \protected@eappto\gls@protected@pagefmts{\, \expandonce{\csname gls#1page\endcsname}}%
  \csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
  \protected@eappto\@wrglossarynumberhook{%
    \noexpand\let\expandonce{\csname org@gls#1\endcsname}%
    \expandonce{\csname#1\endcsname}%
    \noexpand\def\expandonce{\csname#1\endcsname}{%
      \noexpand\@wrglossary@pageformat
      \expandonce{\csname gls#1page\endcsname}%
      \expandonce{\csname org@gls#1\endcsname}%
    }%
  }%
}
\newcommand*{\@wrglossarynumberhook}{}
\newcommand{\@wrglossary@pageformat}[3]{%
  \ifx#3\c@page #1\else #2#3\fi
}
\newcommand*{\@@do@wrglossary}[1]{%
  \ifglsclocations
    \@@do@esc@wrglossary{#1}%
  \else
    \@@do@noesc@wrglossary{#1}%
  \fi
}
\newcommand*{\@@do@noesc@wrglossary}[1]{%
  \expandafter\def\expandafter\@glslocref\expandafter{\theHglentrycounter}%
  \expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
  \ifx\@glsHlocref\@glslocref
    \def\@glo@counterprefix{}%
  \else
    \protected@edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
      {\@glslocref}{\@glsHlocref}%
    }%
    \@do@gls@getcounterprefix
  \fi
  \protected@edef\@gls@label{\glsdetoklabel{#1}}%
  \@@do@@wrglossary
}

```

```

\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodstrue
\newcommand*{\@@do@esc@wrglossary}[1]{% please read documented code!
\beginingroup
\let\gls@orgthe\the
\let\gls@orgnumber\number
\let\gls@orgarabic\@arabic
\let\gls@orgromannumeral\romannumeral
\let\gls@orgalph\@alph
\let\gls@orgAlph\@Alph
\let\gls@orgRoman\@Roman
\ifglswrallowprimitivemods
\def\gls@the##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
\def\the{\expandafter\gls@the}%
\def\gls@number##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
\def\number{\expandafter\gls@number}%
\fi
\def\@arabic##1{%
\ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\romannumeral##1{%
\ifx##1\c@page \gls@romanpage\else\gls@orgromannumeral##1\fi}%
\def\@Roman##1{%
\ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
\ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
\ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\@gls@locref{\theHglentrycounter}%
\endgroup
\@gls@checkmkidxchars\@gls@locref
\expandafter\ifx\theHglentrycounter\theHglentrycounter\relax
\def\@glo@counterprefix{%
\else
\protected@edef\@gls@Hlocref{\theHglentrycounter}%
\@gls@checkmkidxchars\@gls@Hlocref
\edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
{\@gls@locref}{\@gls@Hlocref}%
}%
\@do@gls@getcounterprefix
\fi
\protected@edef\@gls@label{\gls@detoklabel{#1}}%
\@@do@@wrglossary
}
\newcommand*{\@@do@@wrglossary}{%
\ifglxindy
\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)
:locoref \string"\@glo@counterprefix}{\@gls@locoref}\string" %
:attr \string"\@gls@counter\@glo@suffix\string"
\@glo@range
)
}%
\else
\@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@gls@numberformat}%
{\@glo@counterprefix}%
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
\string\glossaryentry{\csname glo@\@gls@label @index\endcsname
\@gls@encapchar\@glo@numfmt}{\@gls@locoref}}%
\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

```

```

\ifglxsindy
  \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]{\glstrytext{#1}}
\newcommand*\gls@save@numberlist}[1]{%
  \ifglssavenumberlist

```

```

\toks@{#1}%
\protected@edef\do@writeaux@info{%
  \noexpand\csgdef{glo@glscurrententrylabel @numberlist}{\the\toks@}%
}%
\@onelevel@sanitize\do@writeaux@info
\protected@write\auxout{}\do@writeaux@info{%
\fi
}
\newcommand*{\warn@noprntglossary}{}%
\ifcsundef{printglossary}{}%
{%
  \@gls@warnonglossdefined
  \undef\printglossary
}
\newcommand*{\@printgloss@checkexists@noignored}[2]{%
  \@ifglossaryexists{#1}%
  {#2}%
  {%
    \ifignoredglossary{#1}%
    {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
    {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
  }%
}
\newcommand*{\@printgloss@checkexists@allowignored}[2]{%
  \s@ifglossaryexists{#1}%
  {#2}%
  {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\newcommand*{\printglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@glossary}%
}
\newcommand*{\printglossaries}{%
  \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}
\newcommand*{\printnoidxglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@noidx@glossary}%
}
\newcommand*{\printnoidxglossaries}{%
  \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
\newcommand*{\@printgloss@setsort}{%
}
\newcommand*{\@gls@preglossaryhook}{%
}
\newcommand*{\@printglossary}[2]{%
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{\csname @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@dotoc@title{\gls@set@toc@title{\@glo@type}}%
\let\@org@glossaryentrynumbers\glossaryentrynumbers
\begin{group}
  \@printgloss@set@sort
  \setkeys{printgloss}{#1}%
  \@printgloss@check@exists{\@glo@type}%
  {%
    \ifx\glossarytitle\@org@glossarytitle
    \else
      \expandafter\let\csname @glo@type@\@glo@type @title\endcsname
        \glossarytitle
    \fi
    \let\currentglossary\@glo@type
    \let\@org@glossaryentrynumbers\glossaryentrynumbers
    \let\gls@nonextpages\@gls@nonextpages
    \let\gls@nextpages\@gls@nextpages
    \let\@nopostdesc\@nopostdesc
    \gls@dotoc@title
    \@glossarystyle
    \let\gls@org@glossaryentryfield\glossentry
    \let\gls@org@glossarysubentryfield\subglossentry
    \renewcommand{\glossentry}[1]{%
      \protected@xdef\gls@currententrylabel{\gls@detoklabel{##1}}%
      \gls@org@glossaryentryfield{##1}%
    }%
    \renewcommand{\subglossentry}[2]{%
      \protected@xdef\gls@currententrylabel{\gls@detoklabel{##2}}%
      \gls@org@glossarysubentryfield{##1}{##2}%
    }%
    \@gls@preglossaryhook
    #2%
  }%
\end{group}
\global\let\glossaryentrynumbers\@org@glossaryentrynumbers
\global\let\warn@noprintglossary\relax
}
\newcommand{\@print@glossary}{%
  \makeatletter
  \@input@{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%
  \IfFileExists{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%

```



```

}%
{\null}%
\ifglxindy
\ifcsundef{@xdy@\@glo@type @language}%
{%
\edef\@do@auxoutstuff{%
\noexpand\AtEndDocument{%
\noexpand\immediate\noexpand\write\@auxout{%
\string\providecommand\string\@xdylanguage[2]{}}%
\noexpand\immediate\noexpand\write\@auxout{%
\string\@xdylanguage{\@glo@type}{\@xdy@main@language}}%
}%
}%
}%
{%
\edef\@do@auxoutstuff{%
\noexpand\AtEndDocument{%
\noexpand\immediate\noexpand\write\@auxout{%
\string\providecommand\string\@xdylanguage[2]{}}%
\noexpand\immediate\noexpand\write\@auxout{%
\string\@xdylanguage{\@glo@type}{\csname @xdy@\@glo@type
@language\endcsname}}%
}%
}%
}%
\@do@auxoutstuff
\edef\@do@auxoutstuff{%
\noexpand\AtEndDocument{%
\noexpand\immediate\noexpand\write\@auxout{%
\string\providecommand\string\@gls@codepage[2]{}}%
\noexpand\immediate\noexpand\write\@auxout{%
\string\@gls@codepage{\@glo@type}{\@gls@codepage}}%
}%
}%
\@do@auxoutstuff
\fi
\renewcommand*{\@warn@nomakeglossaries}{%
\GlossariesWarningNoLine{\string\makeglossaries\space
hasn't been used,^^Jthe glossaries will not be updated}%
}%
}
\newcommand*{\@glo@sortentries}[2]{%
\glosortentrieswarning
\def\@glo@sortinglist{%
\def\@glo@sortinghandler{#1}%
\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@glorder}%
    {%
      \@glo@sortentries{\csuse{\@glo@sorthandler@glorder}}{#1}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\glorder’}{}%
    }%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=standard]}{%}%
  }%
}
\newcommand*{\@glo@sortmacro@case}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@case}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=case]}{%}%
  }%
}
\newcommand*{\@glo@sortmacro@nocase}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
  }%
}

```

```

{%
  \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=nocase]}{}%
}%
}
\newcommand*{\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{}%
  \forglseentries[#1]{\@gls@thislabel}%
  {%
    \xifinlistcs{\@gls@thislabel}{\@glsref@#1}%
    {%
      \listcsadd{\@glo@sortinglist}{\@gls@thislabel}%
    }%
    {%
    }%
  }%
  \cslet{\@glsref@#1}{\@glo@sortinglist}%
}
\newcommand*{\@glo@sortmacro@def@do}[1]{%
  \ifinlistcs{#1}{\@glsref@\@glo@type}%
  {}%
  {%
    \listcsadd{\@glsref@\@glo@type}{#1}%
  }%
  \ifcsdef{\@glo@sortingchildren@#1}%
  {%
    \@glo@addchildren{\@glo@type}{#1}%
  }%
  {}%
}
\newcommand*{\@glo@sortmacro@use}[1]{%
\newcommand*{\@print@noidx@glossary}{%
  \ifcsdef{\@glsref@\@glo@type}%
  {%
    \ifcsdef{\@glo@sortmacro@\@glo@sorttype}%
    {%
      \csuse{\@glo@sortmacro@\@glo@sorttype}{\@glo@type}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}%
    }%
    \glossarysection[\glossarytoctitle]{\glossarytitle}%
    \glossarypreamble
    \def\@gls@currentlettergroup{}%
    \begin{theglossary}%
    \glossaryheader
    \glsresetentrylist
    \forlistcsloop{\@gls@noidx@do}{\@glsref@\@glo@type}%
    \end{theglossary}%
  }%
}

```

```

\glossarypostamble
}%
{%
\@gls@noref@warn{\@glo@type}%
}%
}
\def\glo@grabfirst#1#2\@nil{%
\def\@gls@firsttok{#1}%
\ifdefempty\@gls@firsttok
{%
\def\@glo@thislettergrp{0}%
}%
{%
\@onelevel@sanitize\@gls@firsttok
\expandafter\@glo@grabfirst\@gls@firsttok{}}\@nil
}%
}
\def\@glo@grabfirst#1#2\@nil{%
\ifdefempty\@glo@thislettergrp
{%
\def\@glo@thislettergrp{glssymbols}%
}%
{%
\count@=\ucode‘#1\relax
\ifnum\count@=0\relax
\def\@glo@thislettergrp{glssymbols}%
\else
\ifdefstring\@glo@sorttype{case}%
{%
\count@=‘#1\relax
}%
}%
\edef\@glo@thislettergrp{\the\count@}%
\fi
}%
}
\newcommand{\@gls@noidx@do}[1]{%
\global\letcs{\@gls@loclist}{\glo@\glsdetoklabel{#1}@loclist}%
\ifglshasparent{#1}%
{%
\gls@level=\csuse{\glo@\glsdetoklabel{#1}@level}\relax
\ifdefvoid{\@gls@loclist}
{%
\subglossentry{\gls@level}{#1}{}%
}%
{%
\subglossentry{\gls@level}{#1}%
{%
\glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%

```

```

    }%
  }%
}%
{%
  \letcs{\@gls@sort}{glo@glstetoklabel{#1}@sort}%
  \expandafter\glo@grabfirst\@gls@sort{}\{}@nil
  \ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
  {}%
  {%
    \ifdefempty{\@gls@currentlettergroup}{}%
    {%
      \global\let\@glo@thislettergrp\@glo@thislettergrp
      \glsgroupskip
    }%
    \glsgroupheading{\@glo@thislettergrp}%
  }%
  \global\let\@gls@currentlettergroup\@glo@thislettergrp
  \ifdefvoid{\@gls@loclist}
  {%
    \glossentry{#1}{}%
  }%
  {%
    \glossentry{#1}%
    {%
      \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
    }%
  }%
}%
}
}
\newcommand*{\glsnoidxloclist}[1]{%
  \def\@gls@noidxloclist@sep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxloclisthandler}{#1}%
}
\newcommand*{\glsnoidxloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%
    \@gls@noidxloclist@sep
    #1%
    \def\@gls@noidxloclist@sep{\delimN}%
    \def\@gls@noidxloclist@prev{#1}%
  }%
}
}
\newcommand*{\glsnoidxdisplayloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%

```

```

\@gls@noidxloclist@sep
\@gls@noidxloclist@prev
\def\@gls@noidxloclist@prev{#1}%
}%
}
\newcommand*\glsnoidxdisplayloc[4]{%
\setentrycounter[#1]{#2}%
\csuse{#3}{#4}%
}
\newcommand*\@gls@reference}[3]{%
\glsdoifexistsorwarn{#2}%
{%
\ifcsundef{\@glsref@#1}{\csgdef{\@glsref@#1}{}}{}%
\ifinlistcs{#2}{\@glsref@#1}%
{}%
{\listcsgadd{\@glsref@#1}{#2}}%
\ifcsundef{glo@\glsdetoklabel{#2}@loclist}%
{\csgdef{glo@\glsdetoklabel{#2}@loclist}{}}%
{}%
\listcsgadd{glo@\glsdetoklabel{#2}@loclist}{#3}%
}%
}
\define@key{printgloss}{type}{\def\@glo@type{#1}}
\define@key{printgloss}{title}{%
\def\glossarytitle{#1}%
\let\gls@dotocitle\relax
}
\define@key{printgloss}{toctitle}{%
\def\glossarytoctitle{#1}%
\let\gls@dotocitle\relax
}
\define@key{printgloss}{style}{%
\ifcsundef{\@glsstyle@#1}%
{%
\PackageError{glossaries}%
{Glossary style ‘#1’ undefined}{}%
}%
{%
\def\@glossarystyle{\setglossentrycompatibility
\csname @glsstyle@#1\endcsname}%
}%
}
\define@choicekey{printgloss}{numberedsection}%
[\@gls@numberedsection@val\@gls@numberedsection@nr]%
{false,nolabel,autolabel,nameref}[nolabel]%
{%
\ifcase\@gls@numberedsection@nr\relax
\renewcommand*\@@glossarysecstar{*}%
\renewcommand*\@@glossaryseclabel{}%
\or

```



```

\renewcommand*{\@@glossarysecstar}{}%
\renewcommand*{\@@glossaryseclabel}{}%
\or
\renewcommand*{\@@glossarysecstar}{}%
\renewcommand*{\@@glossaryseclabel}{\label{\glsautoprefix\@glo@type}}%
\or
\renewcommand*{\@@glossarysecstar}{*}%
\renewcommand*{\@@glossaryseclabel}{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}}%
\fi
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
\csuse{glsnogroupskip#1}%
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
\csuse{glsnopostdot#1}%
}
\newcommand*{\GlsEntryCounterLabelPrefix}{glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
\csuse{glsentrycounter#1}%
\@gls@define@glossaryentrycounter
}
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
\csuse{glsesubentrycounter#1}%
\@gls@define@glossarysubentrycounter
}
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
\ifglsnonumberlist
\def\glossaryentrynumbers##1{%
}
\else
\def\glossaryentrynumbers##1{##1}%
\fi}
\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
\newcommand*{\@glo@no@assign@sortkey}[1]{%
\PackageError{glossaries}{‘sort’ key not permitted with
\string\printglossary}%
{The ‘sort’ key may only be used with \string\printnoidxglossary}%
}
\newcommand*{\@glo@assign@sortkey}[1]{%
\def\@glo@sorttype{#1}%
}
\newcommand*{\@glsnonextpages}{%
\gdef\glossaryentrynumbers##1{%
\glsresetentrylist
}%
}
\newcommand*{\@glsnextpages}{%
\gdef\glossaryentrynumbers##1{%
##1\glsresetentrylist}}

```

```

\newcommand*{\glsresetentrylist}{%
  \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
\newcommand*{\glsnonextpages}{}
\newcommand*{\glsnextpages}{}
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
\newcommand*{\glsresetsubentrycounter}{%
  \ifglssubentrycounter
    \setcounter{glossarysubentry}{0}%
  \fi
}
\newcommand*{\glsresetentrycounter}{%
  \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*{\glsesubentryitem}[1]{%
  \ifglsesubentrycounter
    \glsstepsubentry{#1}\glsesubentrycounterlabel
  \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@#1\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][1]{%
  \def\@glo@counterprefix{#1}%
  \ifx\@glo@counterprefix\@empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1}%
  \fi
  \def\glsentrycounter{#2}%
}
\newcommand*{\setglossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \csname @glsstyle@#1\endcsname
  }%
}

```

```

\ifx\@glossary@default@style\relax
  \protected@edef\@glossary@default@style{#1}%
\fi
}
\newcommand*\@glossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \GlossariesWarning
    {Deprecated command \string\glossarystyle.^^J
      I recommend you switch to \string\setglossarystyle\space unless
      you want to maintain backward compatibility}%
    \setglossentrycompatibility
    \csname @glsstyle@#1\endcsname
    \ifcsdef{@glscompstyle@#1}%
    {\setglossentrycompatibility\cuse{@glscompstyle@#1}}%
    {}%
  }%
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\newcommand*\@newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
\newcommand*\@renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
\newcommand*\@glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\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}%
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand*{\acrnameformat}[2]{\acronymfont{#1}}
\newtoks\glskeylisttok
\newtoks\glslabeltok
\newtoks\glsshorttok
\newtoks\gslongtok
\newcommand*{\newacronymhook}{}
\newcommand*{\SetGenericNewAcronym}{%
  \let\@Gls@entryname\@Gls@acentryname
  \renewcommand{\newacronym}[4][]{%
    \ifdefempty{\@glsacronymlists}%
    {%
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
    }%
    {}%
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \gslongtok{##4}%
    \newacronymhook
  }%
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\expandonce{\acronymentry{##2}}},%
      sort={\acronymsort{\the\glsshorttok}{\the\gslongtok}},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\gslongtok},%
      longplural={\the\gslongtok\noexpand\acrpluralsuffix},%
      \GenericAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink{##1}{##2}{\genacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink{##1}{##2}{\Genacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink{##1}{##2}{%
    \mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}%
}

```

```

\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\glsentryfull}[1]{\genacrfullformat{##1}{}}}%
\renewcommand*{\Glsentryfull}[1]{\Genacrfullformat{##1}{}}}%
\renewcommand*{\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}}%
\renewcommand*{\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}}%
}
\newcommand*{\GenericAcronymFields}{description={\the\glslongtok}}
\newcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{#1}}}
\newcommand*{\acronymsort}[2]{#1}
\newcommand*{\setacronymstyle}[1]{%
  \ifcsundef{@glsacr@dispstyle@#1}
  {%
    \PackageError{glossaries}{Undefined acronym style ‘#1’}{}%
  }%
  {%
    \ifdefempty{\@glsacronymlists}%
    {%
      \DeclareAcronymList{\acronymtype}%
    }%
    {}%
    \SetGenericNewAcronym
    \GlsUseAcrStyleDefs{#1}%
    \@for\@gls@type:=\@glsacronymlists\do{%
      \defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDisplayStyle{#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]{%
\glstrylong{##1}##2\space
(\protect\firstacronymfont{\glstryshort{##1}})%
}%
\renewcommand*\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glstryshort{##1}})%
}%
\renewcommand*\genplacrfullformat}[2]{%
\glstrylongpl{##1}##2\space
(\protect\firstacronymfont{\glstryshortpl{##1}})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glstryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glstryshort{##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]{%
\glstrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glstryshort{##1}})%
}%
\renewcommand*\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glstryshort{##1}})%
}%
\renewcommand*\genplacrfullformat}[2]{%
\glstrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glstryshortpl{##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}{\glsacspace}{\glsacspace}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\genacrfullformat}[2]{%
    \protect\firstacronymfont{\glsentryshort{##1}}##2\space
    (\glsentrylong{##1})%
  }%
  \renewcommand*{\Genacrfullformat}[2]{%
    \protect\firstacronymfont{\Glsentryshort{##1}}##2\space
    (\glsentrylong{##1})%
  }%
  \renewcommand*{\genplacrfullformat}[2]{%
    \protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
    (\glsentrylongpl{##1})%
  }%
  \renewcommand*{\Genplacrfullformat}[2]{%
    \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
    (\glsentrylongpl{##1})%
  }%
  \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
  \renewcommand*{\acronymsort}[2]{##1}%
  \renewcommand*{\acronymfont}[1]{##1}%
  \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{%
  \GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
}

```

```

\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{long-sm-short}%
{%
\GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glacrpluralsuffix}%
}
\newacronymstyle{sc-short-long}%
{%
\GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{sm-short-long}%
{%
\GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{%
\GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sp-short-desc}%
{%
\GlsUseAcrEntryDisplayStyle{long-sp-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sp-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glentrylong{##1}\glspacspace{##1}(\acronymfont{\glentryshort{##1}})}%
}

```

```

}
\newacronymstyle{long-sc-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sc-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%

```

```

\GlsUseAcrStyleDefs{sm-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \Glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \mfirstucMakeUppercase
            {\glsentrylongpl{\glslabel}\glsinsert}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \Glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \mfirstucMakeUppercase
          {\glsentrylong{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glsgenentryfmt
  }%
  {\glscustomtext\glsinsert}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\acrfullfmt}[3]{%
    \glslink{##1}{##2}{\glsentrylong{##2}##3\space

```



```

        (\acronymfont{\glentryshort{##2}})}%
\renewcommand*{\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{\Glsentrylong{##2}##3\space
        (\acronymfont{\glentryshort{##2}})}%
\renewcommand*{\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
        \mfirstucMakeUppercase{\glentrylong{##2}##3\space
            (\acronymfont{\glentryshort{##2}})}%
\renewcommand*{\acrfullplfmt}[3]{%
    \glslink[##1]{##2}{\glentrylongpl{##2}##3\space
        (\acronymfont{\glentryshortpl{##2}})}%
\renewcommand*{\Acrfullplfmt}[3]{%
    \glslink[##1]{##2}{%
        \mfirstucMakeUppercase{\glentrylongpl{##2}##3\space
            (\acronymfont{\glentryshortpl{##2}})}%
\renewcommand*{\glentryfull}[1]{%
    \glentrylong{##1}\space(\acronymfont{\glentryshort{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
    \Glsentrylong{##1}\space(\acronymfont{\glentryshort{##1}})%
}%
\renewcommand*{\glentryfullpl}[1]{%
    \glentrylongpl{##1}\space(\acronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
    \Glsentrylongpl{##1}\space(\acronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
    \GlsUseAcrEntryDisplayStyle{dua}%
}%
{%
    \GlsUseAcrStyleDefs{dua}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymentry}[1]{\acronymfont{\glentrylong{##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{\glentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}%
\newacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glacrpluralsuffix}%
}%
\newacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*{\GenericAcronymFields}{}%
}

```

```

\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newcommand*{\DefineAcronymSynonyms}{%
  \let\acs\acrshort
  \let\Acs\Acrshort
  \let\acsp\acrshortpl
  \let\Acsp\Acrshortpl
  \let\acl\acrlong
  \let\Acl\Acrlong
  \let\aclp\acrlongpl
  \let\Aclp\Acrlongpl
  \let\acf\acrfull
  \let\Acfp\acrfullpl
  \let\Acf\Acrfull
  \let\Acfp\Acrfullpl
  \let\ac\gls
  \let\Ac\Gls
  \let\acp\glspl
  \let\Acp\Glspl
}
\ifglsacrshortcuts
  \DefineAcronymSynonyms
\fi
\newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsentryfmt}%
}
\newcommand*{\DefaultNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      sort={\the\glsshorttok},%
      text={\the\glsshorttok},%
      first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
      plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      firstplural={\acrfullformat{\noexpand\expandonce\noexpand\@glo@longpl}%
        {\noexpand\expandonce\noexpand\@glo@shortpl}},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      \the\glskeylisttok
    }%
  }%
  \let\@org@gls@assign@firstpl\gls@assign@firstpl

```

```

\let\@org@gl@assign@plural\gl@assign@plural
\let\@org@gl@assign@descplural\gl@assign@descplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@gl@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\@org@gl@assign@firstpl
\let\gl@assign@plural\@org@gl@assign@plural
\let\gl@assign@symbolplural\@org@gl@assign@symbolplural
\@gl@noexpand@assign@default@plurals
}
\newcommand*{\@gl@noexpand@assign@default@plurals}{%
  \expandafter\ifx\csname glo@the\glslabeltok @plural\endcsname\@gl@@default@acr@plural
    \csxdef{glo@the\glslabeltok @plural}{\expandonce\@glo@shortpl}%
  \fi
  \expandafter\ifx\csname glo@the\glslabeltok @firstpl\endcsname\@gl@@default@acr@firstplural
    \csxdef{glo@the\glslabeltok @firstpl}{%
      \expandonce\@glo@longpl\glsspace(\expandonce\@glo@shortpl)}%
  \fi
  \expandafter\ifx\csname glo@the\glslabeltok @descplural\endcsname\@gl@@default@acr@descplural
    \csxdef{glo@the\glslabeltok @descplural}{\expandonce\@glo@longpl}%
  \fi
}
\newcommand*{\@gl@@default@acr@plural}{\expandonce\@glo@shortpl}
\newcommand*{\@gl@@default@acr@firstplural}{%
  \expandonce\@glo@longpl\glsspace(\expandonce\@glo@shortpl)%
}
\newcommand*{\@gl@@default@acr@descplural}{\expandonce\@glo@longpl}
\newcommand*{\SetDefaultAcronymStyle}{%
  \@for\@gl@type:=\@gl@acronymlists\do{%
    \SetDefaultAcronymDisplayStyle{\@gl@type}%
  }%
}
\renewcommand{\newacronym}[4][{}]{%
  \ifx\@gl@acronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetDefaultAcronymDisplayStyle{\@glo@type}%
  \fi
  \gl@keylisttok{##1}%
  \gl@labeltok{##2}%
  \gl@shorttok{##3}%
  \gl@longtok{##4}%
  \newacronymhook

```

```

\DefaultNewAcronymDef
}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
\newcommand*{\acrlinkfootnote}[3]{%
  \footnote{\glslink{#1}{#2}{#3}}%
}
\newcommand*{\acrnolinkfootnote}[3]{%
  \footnote{#3}%
}
\newcommand*{\SetDescriptionFootnoteAcronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{%
    \ifdefempty\glscustomtext
    {%
      \ifglsused{\glslabel}%
      {%
        \acronymfont{\glsentryfmt}%
      }%
      {%
        \firstacronymfont{\glsentryfmt}%
        \ifgls hassymbol{\glslabel}%
        {%
          \expandafter\protect\expandafter\acrfootnote\expandafter
            {\@gls@link@opts}{\@gls@link@label}%
          {%
            \glsifplural
              {\glsentrysymbolplural{\glslabel}}%
              {\glsentrysymbol{\glslabel}}%
            }%
          }%
        }%
      }%
    }%
    {\glscustomtext\glsinsert}%
  }%
}
\newcommand*{\DescriptionFootnoteNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\noexpand\acronymfont{\the\glsshorttok}},%
      sort={\the\glsshorttok},%
      first={\the\glsshorttok},%
      firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      text={\the\glsshorttok},%
      plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
    }%
  }%
}

```

```

        longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
        symbol={\the\glslongtok},%
        symbolplural={\noexpand\expandonce\noexpand\@glo@longpl},%
        \the\glskeylisttok
    }%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
    \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
    \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
    \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionFootnoteAcronymStyle}{%
    \ifglsacrsmallcaps
        \@gls@deprecated@acr@warning{footnote,smallcaps,description}%
        {footnote-sc-desc}{short-sc-footnote-desc}%
    \else
        \ifglsacrsmaller
            \@gls@deprecated@acr@warning{footnote,smaller,description}%
            {footnote-sm-desc}{short-sm-footnote-desc}%
        \else
            \@gls@deprecated@acr@warning{footnote,description}%
            {footnote-desc}{short-footnote-desc}%
        \fi
    \fi
\renewcommand{\newacronym}[4][{}]{%
    \ifx\@glsacronymlists\@empty
        \def\@glo@type{\acronymtype}%
        \setkeys{glossentry}{##1}%
        \DeclareAcronymList{\@glo@type}%
        \SetDescriptionFootnoteAcronymDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \DescriptionFootnoteNewAcronymDef
}%

```

```

\@for\@gls@type:=\@glsacronymlists\do{%
  \SetDescriptionFootnoteAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
  \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
\else
  \ifglsacrsmaller
    \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
  \fi
\fi
\ifglsacrdua
  \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
can’t both be set}{}%
\fi
}%
\newcommand*{\SetDescriptionDUAAcronymDisplayStyle}[1]{%
  \defglsentryfmt[##1]{\glsentryfmt}%
}
\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

```

```

\ifglused{\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]{%
  \def\glsentryfmt[#1]{%
    \ifdefempty\gls@customtext
    {%
      \let\gls@org@insert\glsinsert
      \let\glsinsert\@empty
      \ifglsused{\glslabel}%
      {%
        \acronymfont{\gls@genentryfmt}\gls@org@insert
      }%
      {%
        \gls@genentryfmt
        \ifgls@hasymbol{\glslabel}%
        {%
          \gls@ifplural
          {%
            \def\@glo@symbol{\gls@entrysymbolplural{\glslabel}}%
          }%
          {%
            \def\@glo@symbol{\gls@entrysymbol{\glslabel}}%
          }%
          \space
          (\gls@scapscase
          {\firstacronymfont{\@glo@symbol}}%
          {\firstacronymfont{\@glo@symbol}}%
          {\firstacronymfont{\mfirstucMakeUppercase{\@glo@symbol}}})%
        }%
        {}%
      }%
    }%
    {\gls@customtext\glsinsert}%
  }%
}
\newcommand*{\SmallNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\noexpand\acronymfont{\the\gls@shorttok}},%
      sort={\the\gls@shorttok},%
    }
  }

```

```

text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
first={\the\glslongtok},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\noexpand\@glo@first},%
descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
symbol={\the\glsshorttok},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@@gls@expand@field{##1}{descplural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@descplural\@org@gls@assign@descplural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetSmallAcronymStyle}{%
  \ifglsacrsmallcaps
    \@gls@deprecated@acr@warning{smallcaps}{long-sc-short}{long-short-sc}%
  \else
    \@gls@deprecated@acr@warning{smaller}{long-sm-short}{long-short-sm}%
  \fi
\renewcommand{\newacronym}[4][\]{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetSmallAcronymDisplayStyle{\@glo@type}%
  \fi

```

```

\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\SmallNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
  \SetSmallAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
  \renewcommand*\acronymfont}[1]{\textsc{##1}}
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
\else
  \renewcommand*\acronymfont}[1]{\textsmaller{##1}}
\fi
\ifglsacrdua
  \ifglsacrsmallcaps
    \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
      can’t both be set}{}%
  \else
    \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
      can’t both be set}{}%
  \fi
\fi
}%
\newcommand*\SetDUADisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsentryfmt}%
}
\newcommand*\DUANewAcronymDef{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      text={\the\glslongtok},%
      first={\the\glslongtok},%
      plural={\noexpand\expandonce\noexpand\@glo@longpl},%
      firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      symbol={\the\glsshorttok},%
      symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      \the\glskeylisttok
    }%
  }%
}%

```



```

\let\@org@gl@s@assign@firstpl\gl@s@assign@firstpl
\let\@org@gl@s@assign@plural\gl@s@assign@plural
\let\@org@gl@s@assign@symbolplural\gl@s@assign@symbolplural
\let\@org@gl@s@assign@descplural\gl@s@assign@descplural
\def\gl@s@assign@firstpl##1##2{%
  \@@gl@s@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@s@assign@plural##1##2{%
  \@@gl@s@expand@field{##1}{plural}{##2}%
}%
\def\gl@s@assign@symbolplural##1##2{%
  \@@gl@s@expand@field{##1}{symbolplural}{##2}%
}%
\def\gl@s@assign@descplural##1##2{%
  \@@gl@s@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@s@assign@firstpl\@org@gl@s@assign@firstpl
\let\gl@s@assign@plural\@org@gl@s@assign@plural
\let\gl@s@assign@symbolplural\@org@gl@s@assign@symbolplural
\let\gl@s@assign@descplural\@org@gl@s@assign@descplural
}
\newcommand*{\SetDUASStyle}{%
  \@gl@s@deprecated@acr@warning{dua}{dua}{long-noshort}%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@gl@sacronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetDUADisplayStyle{\@glo@type}%
    \fi
    \gl@keylisttok{##1}%
    \gl@labeltok{##2}%
    \gl@shorttok{##3}%
    \gl@longtok{##4}%
    \newacronymhook
    \DUANewAcronymDef
  }%
  \@for\@gl@s@type:=\@gl@sacronymlists\do{%
    \SetDUADisplayStyle{\@gl@s@type}%
  }%
}
\newcommand*{\@gl@s@deprecated@acr@warning}[3]{%
  \GlossariesWarning{Acronym style option #1 deprecated.
    Use \string\setacronymstyle{#2} with just glossaries.sty or
    use \string\setabbreviationstyle[acronym]{#3} with
    glossaries-extra.sty}%
}
\newcommand*{\SetAcronymStyle}{%
  \SetDefaultAcronymStyle

```

```

\ifglsacrdescription
\ifglsacrfootnote
\SetDescriptionFootnoteAcronymStyle
\else
\ifglsacrdua
\SetDescriptionDUAAcronymStyle
\else
\SetDescriptionAcronymStyle
\fi
\fi
\else
\ifglsacrfootnote
\SetFootnoteAcronymStyle
\else
\ifthenelse{\boolean{glsacrsmallcaps}\OR
\boolean{glsacrsmaller}}{%
}%
\SetSmallAcronymStyle
}%
}%
\ifglsacrdua
\SetDUASyle
\fi
}%
\fi
\fi
}
\SetAcronymStyle
\newcommand*{\SetCustomDisplayStyle}[1]{%
\defglsentryfmt[#1]{\glsentryfmt}%
}
\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*{\showgloftype}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@type\endcsname
}
\newcommand*{\showglocounter}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@counter\endcsname
}
\newcommand*{\showglouserii}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@userii\endcsname
}
\newcommand*{\showglouseriii}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@useriii\endcsname
}
\newcommand*{\showglouseriv}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@useriv\endcsname
}
\newcommand*{\showglouserv}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@userv\endcsname
}
\newcommand*{\showglouservi}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@uservi\endcsname
}
\newcommand*{\showgloname}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@name\endcsname
}
\newcommand*{\showglodesc}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@desc\endcsname
}
\newcommand*{\showglodescplural}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@descplural\endcsname
}
\newcommand*{\showglosort}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@sort\endcsname
}
\newcommand*{\showglosymbol}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@symbol\endcsname
}
\newcommand*{\showglosymbolplural}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolplural\endcsname
}
\newcommand*{\showgloshort}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@short\endcsname
}

```

```

}
\newcommand*{\showglolong}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname
}
\newcommand*{\showgloindex}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname
}
\newcommand*{\showgloflag}[1]{%
  \expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname
}
\newcommand*{\showgloclist}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@loclist\endcsname
}
\newcommand*{\showglofield}[2]{%
  \cshow{glo@\glsdetoklabel{#1}@#2}%
}
\newcommand*{\showacronymlists}{%
  \show\@glsacronymlists
}
\newcommand*{\showglossaries}{%
  \show\@glo@types
}
\newcommand*{\showglossaryin}[1]{%
  \expandafter\show\csname @glo@type@#1@in\endcsname
}
\newcommand*{\showglossaryout}[1]{%
  \expandafter\show\csname @glo@type@#1@out\endcsname
}
\newcommand*{\showglossarytitle}[1]{%
  \expandafter\show\csname @glo@type@#1@title\endcsname
}
\newcommand*{\showglossarycounter}[1]{%
  \expandafter\show\csname @glo@type@#1@counter\endcsname
}
\newcommand*{\showglossaryentries}[1]{%
  \expandafter\show\csname glist@#1\endcsname
}
\csname ifglscompatible-2.07\endcsname
  \RequirePackage{glossaries-compatible-207}[v4.49]
\fi

```

8.19 Rollback v4.49 (glossaries-accsupp-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-accsupp}[2021/11/01 v4.49 (NLCT)
  Experimental glossaries accessibility]
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
\ProcessOptions
\@ifpackageloaded{glossaries-extra}

```

```

{%
\ifx\@glsxtr@doaccsupp\empty
\GlossariesWarning{The 'glossaries-accsupp'
package has been loaded\MessageBreak
after the 'glossaries-extra' package. This\MessageBreak
can cause a failure to integrate both packages. \MessageBreak
Either use the 'accsupp' option when you load\MessageBreak
'glossaries-extra' or load 'glossaries-accsupp'\MessageBreak
before loading 'glossaries-extra'}%
\fi
}
{}
\def\compatibleglossentry#1#2{%
\toks@{#2}%
\protected@edef\@do@glossentry{%
\noexpand\accsuppglossaryentryfield{#1}%
{\noexpand\glsnamefont
{\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@name\endcsname}}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@desc\endcsname}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@symbol\endcsname}%
{\the\toks@}%
}%
\@do@glossentry
}
\def\compatiblesubglossentry#1#2#3{%
\toks@{#3}%
\protected@edef\@do@subglossentry{%
\noexpand\accsuppglossarysubentryfield{\number#1}%
{#2}%
{\noexpand\glsnamefont
{\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@name\endcsname}}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@desc\endcsname}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@symbol\endcsname}%
{\the\toks@}%
}%
\@do@subglossentry
}
\RequirePackage{glossaries}[v4.49]
\providecommand{\gls@accsupp@engine}{accsupp}
\providecommand{\gls@accessibility}[4]{#4}
\ifdefstring\gls@accsupp@engine{accsupp}
{
\RequirePackage{accsupp}
\renewcommand{\gls@accessibility}[4]{%
\BeginAccSupp{#1,#2={#3}}#4\EndAccSupp{}}%
}
}
{}
\newcommand{\gls@accessibility}[4][]{%
\@gls@showaccsupp{#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}{useriiaaccess},%
  {user3access}{useriiiaaccess},%
  {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@useriiaaccess{\relax}%
  \def\@glo@useriiiaaccess{\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{glstr\glscategory{#4}#3accsupp}%
    {\csname glstr\glscategory{#4}#3accsupp\endcsname{#1}{#2}}%
    {%
      \ifcsdef{glstr\glscategory{#4}accsupp}%
      {\csname glstr\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@glstetoklabel{#2}@descaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{desc}{#1}{\glsentrydescaccess{#2}}}%
}
\newrobustcmd*{\glsdescriptionpluralaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@descpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{descplural}{#1}{\glsentrydescpluralaccess{#2}}}%
}
\newrobustcmd*{\glsshortaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@shortaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{short}{#1}{\glsentryshortaccess{#2}}}%
}
\newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@shortpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glsentryshortpluralaccess{#2}}}%
}
\newrobustcmd*{\glslongaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@longaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{long}{#1}{\glsentrylongaccess{#2}}}%
}
\newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@longpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{longpl}{#1}{\glsentrylongpluralaccess{#2}}}%
}
\newrobustcmd*{\glsuseriaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@useriaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useri}{#1}{\glsentryuseriaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@useriiaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{userii}{#1}{\glsentryuseriiaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiiaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@useriiiaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useriii}{#1}{\glsentryuseriiiaccess{#2}}}%
}
\newrobustcmd*{\glsuserivaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@userivaccess}%

```

```

    {#1}%
    {\@gls@fieldaccess@display{#2}{useriv}{#1}{\glsentryuserivaccess{#2}}}%
  }
\newrobustcmd*{\glsuservaccessdisplay}[2]{%
  \ifcsundef{gls@glstoklabel{#2}@uservaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{userv}{#1}{\glsentryuservaccess{#2}}}%
}
\newrobustcmd*{\glsuserviaccessdisplay}[2]{%
  \ifcsundef{gls@glstoklabel{#2}@userviaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{uservi}{#1}{\glsentryuserviaccess{#2}}}%
}
\newrobustcmd*{\glsaccessdisplay}[3]{%
  \ifcsundef{gls#1accessdisplay}%
  {%
    \PackageError{glossaries-accsupp}{No accessibility support
      for key ‘#1’}{}%
  }%
  {%
    \csname gls#1accessdisplay\endcsname{#2}{#3}%
  }%
}
\renewcommand*{\@gls@default@entryfmt}[2]{%
  \ifdefempty{glscustomtext}
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused{glslabel}
        {%
          #2{\glspluralaccessdisplay
            {\glsentryplural{\glslabel}}{\glslabel}}%
          {\glsdescriptionpluralaccessdisplay
            {\glsentrydescplural{\glslabel}}{\glslabel}}%
          {\glsymbolpluralaccessdisplay
            {\glsentrysymbolplural{\glslabel}}{\glslabel}}
          {\glsinsert}%
        }%
      }%
      #1{\glsfirstpluralaccessdisplay
        {\glsentryfirstplural{\glslabel}}{\glslabel}}%
        {\glsdescriptionpluralaccessdisplay
          {\glsentrydescplural{\glslabel}}{\glslabel}}%
          {\glsymbolpluralaccessdisplay
            {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
          {\glsinsert}%
        }%
      }%
    }%
  }%
}

```

```

{%
  \ifglused\glslabel
  {%
    #2{\glspluralaccessdisplay
      {\Glsentryplural{\glslabel}}{\glslabel}}%
    {\glsdescriptionpluralaccessdisplay
      {\glsentrydescplural{\glslabel}}{\glslabel}}%
    {\glssymbolpluralaccessdisplay
      {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
  {%
    #1{\glsfirstpluralaccessdisplay
      {\Glsentryfirstplural{\glslabel}}{\glslabel}}%
    {\glsdescriptionpluralaccessdisplay
      {\glsentrydescplural{\glslabel}}{\glslabel}}%
    {\glssymbolpluralaccessdisplay
      {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
}%
{%
  \ifglused\glslabel
  {%
    \MakeUppercase{%
      #2{\glspluralaccessdisplay
        {\glsentryplural{\glslabel}}{\glslabel}}%
        {\glsdescriptionpluralaccessdisplay
          {\glsentrydescplural{\glslabel}}{\glslabel}}%
        {\glssymbolpluralaccessdisplay
          {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
        {\glsinsert}}}%
    }%
    {%
      \MakeUppercase{%
        #1{\glsfirstpluralaccessdisplay
          {\glsentryfirstplural{\glslabel}}{\glslabel}}%
          {\glsdescriptionpluralaccessdisplay
            {\glsentrydescplural{\glslabel}}{\glslabel}}%
          {\glssymbolpluralaccessdisplay
            {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
          {\glsinsert}}}%
        }%
      }%
    }%
  }%
  \glscapscase
  {%
    \ifglused\glslabel
    {%

```

```

#2{\glstextaccessdisplay
  {\glentrytext{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glentrydesc{\glslabel}}{\glslabel}}%
  {\glssymbolaccessdisplay
    {\glentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
{%
#1{\glsfirstaccessdisplay
  {\glentryfirst{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glentrydesc{\glslabel}}{\glslabel}}%
  {\glssymbolaccessdisplay
    {\glentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
}%
{%
\ifglsused\glslabel
{%
#2{\glstextaccessdisplay
  {\glentrytext{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glentrydesc{\glslabel}}{\glslabel}}%
  {\glssymbolaccessdisplay
    {\glentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
{%
#1{\glsfirstaccessdisplay
  {\glentryfirst{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glentrydesc{\glslabel}}{\glslabel}}%
  {\glssymbolaccessdisplay
    {\glentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
}%
{%
\ifglsused\glslabel
{%
\MakeUppercase{%
#2{\glstextaccessdisplay
  {\glentrytext{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glentrydesc{\glslabel}}{\glslabel}}%
  {\glssymbolaccessdisplay
    {\glentrysymbol{\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@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glsshortaccessdisplay
{\MakeUppercase{\glentryshort{#2}}}{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@acrlong#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glslongaccessdisplay{\glentrylong{#2}}{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@Acrlong#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glslongaccessdisplay{\Glsentrylong{#2}}{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@ACRlong#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax

```

```

\let\glsifplural\@secondoftwo
\let\glsupcase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \acronymfont{\glslongaccessdisplay{%
    \MakeUppercase{\glsentrylong{#2}}}{#2}#3}%
  }%
  \@gls@link{#1}{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@gls@text@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@text@accessdisplay{\glsentrytext{#2}}{#2}#3}%
}
\def\@Gls@text@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@text@accessdisplay{\Glsentrytext{#2}}{#2}#3}%
}
\def\@GLStext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\gls@text@accessdisplay{\mfirstucMakeUppercase{\glsentrytext{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@gls@first@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@first@accessdisplay{\glsentryfirst{#2}}{#2}#3}%
}
\def\@Gls@first@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@first@accessdisplay{\Glsentryfirst{#2}}{#2}#3}%
}
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\gls@first@accessdisplay{\mfirstucMakeUppercase{\glsentryfirst{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@gls@plural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@plural@accessdisplay{\glsentryplural{#2}}{#2}#3}%
}
\def\@Gls@plural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@plural@accessdisplay{\Glsentryplural{#2}}{#2}#3}%
}
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\gls@plural@accessdisplay{\mfirstucMakeUppercase{\glsentryplural{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@gls@firstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@firstplural@accessdisplay{\glsentryfirstplural{#2}}{#2}#3}%
}
\def\@Gls@firstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@firstplural@accessdisplay{\Glsentryfirstplural{#2}}{#2}#3}%
}

```

```

\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsfirstpluralaccessdisplay{\mfirstucMakeUppercase{\glsentryfirstplural{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glsentryname{#2}}{#2}#3}%
}
\def\@Glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\Glsentryname{#2}}{#2}#3}%
}
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsnameaccessdisplay{\mfirstucMakeUppercase{\glsentryname{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\glsentrydesc{#2}}{#2}#3}%
}
\def\@Glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
}
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsdescriptionaccessdisplay{\mfirstucMakeUppercase{\glsentrydesc{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\glsentrydescplural{#2}}{#2}#3}%
}
\def\@Glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
}
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsdescriptionpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrydescplural{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
}
\def\@Glsymbol@#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{\glentrysymbolplural{#2}}{#2}#3}%
}
\def\@Glsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\glentrysymbolplural{#2}}{#2}#3}%
}
\def\@GLSsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{%
\glssymbolpluralaccessdisplay{\mfirstucMakeUppercase{\glentrysymbolplural{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@glsuseri@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glentryuseri{#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{\glentryuseri{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@glsuserii@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glentryuserii{#2}}{#2}#3}%
}
\def\@Glsuser@ii#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glsentryuserii{#2}}{#2}#3}%
}
\def\@GLSuserii@#1#2[#3]{%
\@gls@field@link{#1}{#2}{%
\glsuseriiaccessdisplay{\mfirstucMakeUppercase{\glentryuserii{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@glsuseriii@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\glentryuseriii{#2}}{#2}#3}%
}
\def\@Glsuser@iii#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glsentryuseriii{#2}}{#2}#3}%
}
\def\@GLSuseriii@#1#2[#3]{%
\@gls@field@link{#1}{#2}{%
\glsuseriiiaccessdisplay{\mfirstucMakeUppercase{\glentryuseriii{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@glsuseriv@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glentryuseriv{#2}}{#2}#3}%
}
\def\@Glsuser@iv#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
}
\def\@GLSuseriv@#1#2[#3]{%

```

```

\@gls@field@link{#1}{#2}%
{\glsuserivaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriv{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@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{\glentryshort{##1}}}{##1}}
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\renewacronymstyle{short-long}%
{%
  \ifglshaslong{\glslabel}{\glsгенacfmt}{\glsгенentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
  \glsshortaccessdisplay
    {\protect\firstacronymfont{\glentryshort{##1}}}{##1}##2\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
  \glsshortaccessdisplay
    {\protect\firstacronymfont{\Glsentryshort{##1}}}{##1}##2\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
  \glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\glentryshortpl{##1}}}{##1}##2\space
  (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2\space
  (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\renewacronymstyle{long-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%

```

```

\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}))%
}
\renewacronymstyle{long-sc-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sc-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}))%
}
\renewacronymstyle{long-sm-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sm-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}))%
}
\renewacronymstyle{short-long-desc}%
{%
\GlsUseAcrEntryDispStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}))%
}
\renewacronymstyle{sc-short-long-desc}%
{%
\GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
\GlsUseAcrStyleDefs{sc-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}))%
}

```

```

}
\renewacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
  }
\renewacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glslongpluralaccessdisplay{\glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay{\Glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay
            {\mfirstucMakeUppercase{\glsentrylongpl{\glslabel}}}{\glslabel}%
            {\mfirstucMakeUppercase{\glsinsert}}%
        }%
      }%
      {%
        \glscapscase
        {%
          \glslongaccessdisplay{\glsentrylong{\glslabel}}{\glslabel}\glsinsert
        }%
        {%
          \glslongaccessdisplay{\Glsentrylong{\glslabel}}{\glslabel}\glsinsert
        }%
        {%
          \glslongaccessdisplay
            {\mfirstucMakeUppercase
              {\glsentrylong{\glslabel}\glsinsert}}{\glslabel}%
            {\mfirstucMakeUppercase{\glsinsert}}%
        }%
      }%
    }%
  }%
}

```

```

    }%
  }%
  {%
    \glsgenentryfmt
  }%
}%
{\glscustomtext\glsinsert}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay
      {\mfirstucMakeUppercase{\glsentrylong{##2}}{##2}##3\space
      (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\glsentrylongpl{##2}}{##2}##3\space
      (\glsshortpluralaccessdisplay
        {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\Glsentrylongpl{##2}}{##2}##3\space
      (\glsshortpluralaccessdisplay
        {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\mfirstucMakeUppercase{\glsentrylongpl{##2}}{##2}##3\space
      (\glsshortpluralaccessdisplay
        {\acronymfont{\glsentryshortpl{##2}}}{##2})}}}%
\renewcommand*{\glsentryfull}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\glsentryfullpl}[1]{%

```

```

\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}\space
(\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
\glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space
(\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\renewacronymstyle{dua-desc}%
{%
\GlsUseAcrEntryDispStyle{dua}%
}%
{%
\GlsUseAcrStyleDefs{dua}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\acronymfont{\glsentrylong{##1}}}{##1}%
\renewcommand*{\acronymsort}[2]{##2}%
}%
\renewacronymstyle{footnote}%
{%
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\glsshortaccessdisplay
{\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}}%
}

```



```

}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##2}}}{##2}##3\space
    (\gslongaccessdisplay{\glentrylong{##2}}{##2}))}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##2}}}{##2}##3\space
    (\gslongaccessdisplay{\glentrylong{##2}}{##2}))}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortaccessdisplay
      {\mfirstucMakeUppercase
        {\acronymfont{\glentryshort{##2}}}{##2}##3\space
        (\gslongaccessdisplay{\glentrylong{##2}}{##2}))}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\glentryshortpl{##2}}}{##2}##3\space
      (\gslongpluralaccessdisplay{\glentrylongpl{##2}}{##2}))}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\glentryshortpl{##2}}}{##2}##3\space
      (\gslongpluralaccessdisplay{\glentrylongpl{##2}}{##2}))}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\mfirstucMakeUppercase
        {\acronymfont{\glentryshortpl{##2}}}{##2}##3\space
        (\gslongpluralaccessdisplay{\glentrylongpl{##2}}{##2}))}}%
\renewcommand*{\glentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}\space
  (\gslongaccessdisplay{\glentrylong{##1}}{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}\space
  (\gslongaccessdisplay{\glentrylong{##1}}{##1})}%
\renewcommand*{\glentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\glentryshortpl{##1}}}{##1}\space
    (\gslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\glentryshortpl{##1}}}{##1}\space
    (\gslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})}%

```

```

}
\renewacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glstextup{\glspluralsuffix}}%
}%
\renewacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}%
\renewacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}

```

```

}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
  }
  \newcommand*{\glsdefaultshortaccess}[2]{#1}
  \renewcommand*{\newacronymhook}{%
    \protected@edef\@gls@keylist{%
      shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
      shortpluralaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
      \the\glskeylisttok}%
    \expandafter\glskeylisttok\expandafter{\@gls@keylist}%
  }
  \renewcommand*{\DefaultNewAcronymDef}{%
    \edef\@do@newglossaryentry{%
      \noexpand\newglossaryentry{\the\glslabeltok}%
      {%
        type=\acronymtype,%
        name={\the\glsshorttok},%
        description={\the\glslongtok},%
        descriptionaccess=\relax,%
        text={\the\glsshorttok},%
        access={\noexpand\@glo@textaccess},%
        sort={\the\glsshorttok},%
        short={\the\glsshorttok},%
        shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
        shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
        long={\the\glslongtok},%
        longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
        descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
        first={\noexpand\glslongaccessdisplay
          {\the\glslongtok}{\the\glslabeltok}\space
          (\noexpand\glsshortaccessdisplay
            {\the\glsshorttok}{\the\glslabeltok})},%
        plural={\the\glsshorttok\acrpluralsuffix},%
        firstplural={\noexpand\glslongpluralaccessdisplay
          {\noexpand\@glo@longpl}{\the\glslabeltok}\space
          (\noexpand\glsshortpluralaccessdisplay
            {\noexpand\@glo@shortpl}{\the\glslabeltok})},%
        firstaccess=\relax,%
        firstpluralaccess=\relax,%
        textaccess={\noexpand\@glo@shortaccess},%
        \the\glskeylisttok
      }%
    }%
  }
  \let\@org@gls@assign@firstpl\gls@assign@firstpl

```

```

\let\@org@gl@assign@plural\gl@assign@plural
\let\@org@gl@assign@descplural\gl@assign@descplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@gl@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\@org@gl@assign@firstpl
\let\gl@assign@plural\@org@gl@assign@plural
\let\gl@assign@symbolplural\@org@gl@assign@symbolplural
\@gl@noexpand@assign@default@plurals
}
\renewcommand*{\DescriptionFootnoteNewAcronymDef}{%
  \edef\do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\gl@labeltok}%
    {%
      type=\acronymtype,%
      name={\noexpand\acronymfont{\the\gl@shorttok}},%
      sort={\the\gl@shorttok},%
      text={\the\gl@shorttok},%
      short={\the\gl@shorttok},%
      shortplural={\the\gl@shorttok\noexpand\acrpluralsuffix},%
      shortaccess={\gl@defaultshortaccess{\the\gl@longtok}{\the\gl@shorttok}},%
      long={\the\gl@longtok},%
      longplural={\the\gl@longtok\noexpand\acrpluralsuffix},%
      access={\noexpand\@glo@textaccess},%
      plural={\the\gl@shorttok\noexpand\acrpluralsuffix},%
      symbol={\the\gl@longtok},%
      symbolplural={\the\gl@longtok\noexpand\acrpluralsuffix},%
      firstpluralaccess=\relax,
      textaccess={\noexpand\@glo@shortaccess},%
      \the\gl@keylisttok
    }%
  }%
\let\@org@gl@assign@firstpl\gl@assign@firstpl
\let\@org@gl@assign@plural\gl@assign@plural
\let\@org@gl@assign@symbolplural\gl@assign@symbolplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@symbolplural##1##2{%
  \@gl@expand@field{##1}{symbolplural}{##2}%
}%

```

```

}%
\@do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\DescriptionNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand
\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*{\showglongnameaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@access\endcsname
}
\newcommand*{\showgloptextaccess}[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)]
\ifglxsindy
  \renewcommand*\GlsAddXdyAttribute[1]{%
    \edef\xdyattributes{\@xdyattributes ^^J \string"#1\string"}%
    \expandafter\toks@\expandafter{\@xdylocref}%
    \edef\xdylocref{\the\toks@ ^^J}%
    (markup-locref
    :open \string"\string~\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{glshypernumber}
\GlsAddXdyAttribute{textrm}
\GlsAddXdyAttribute{textsf}
\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glshypernumber}
\GlsAddXdyAttribute{hyperm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hyperit}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}

```

```

\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}
\ifglxsindy
  \renewcommand*{\GlsAddXdyLocation}[2]{%
    \edef\xdyuserlocationdefs{%
      \xdyuserlocationdefs ^^J%
      (define-location-class \string"#1\string"^^J\space\space
        \space(#2))
    }%
    \edef\xdyuserlocationnames{%
      \xdyuserlocationnames^^J\space\space\space
      \string"#1\string"%
    }
  }
\fi
\renewcommand{\@do@wrglossary}[1]{%
\ifglxsindy
  \expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
  \def\@glo@range{}%
  \expandafter\if\@glo@prefix(\relax
    \def\@glo@range{:open-range}%
  \else
    \expandafter\if\@glo@prefix\relax
      \def\@glo@range{:close-range}%
    \fi
  \fi
  \protected@edef\@glslocref{\thegl@sentrycounter}%
  \@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}{\thegl@sentrycounter}}%
  \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%
}

```

```

\ifglxindy
\def\writeist{%
  \openout\glswrite=\istfilename
  \write\glswrite{;; xindy style file created by the glossaries
    package in compatible-2.07 mode}%
  \write\glswrite{;; for document '\jobname' on
    \the\year-\the\month-\the\day}%
  \write\glswrite{^^J; required styles^^J}
  \for\@xdystyle:=\@xdyrequiredstyles\do{%
    \ifx\@xdystyle\@empty
    \else
      \protected@write\glswrite{{(require
        \string"\@xdystyle.xdy\string")}}%
    \fi
  }%
  \write\glswrite{^^J%
    ; list of allowed attributes (number formats)^^J}%
  \write\glswrite{(define-attributes ((\@xdyattributes)))}%
  \write\glswrite{^^J; user defined alphabets^^J}%
  \write\glswrite{\@xdyuseralphabets}%
  \write\glswrite{^^J; location class definitions^^J}%
  \protected@edef\@gls@roman{\@roman{0}\string"
    \string"roman-numbers-lowercase\string" :sep \string"}}%
  \@onelevel@sanitize\@gls@roman
  \edef\@tmp{\string" \string"roman-numbers-lowercase\string"
    :sep \string"}%
  \@onelevel@sanitize\@tmp
  \ifx\@tmp\@gls@roman
    \write\glswrite{(define-location-class
      \string"roman-page-numbers\string"^^J\space\space\space
      (\string"roman-numbers-lowercase\string")
      :min-range-length \@glsminrange)}}%
  \else
    \write\glswrite{(define-location-class
      \string"roman-page-numbers\string"^^J\space\space\space
      (:sep "\@gls@roman")
      :min-range-length \@glsminrange)}}%
  \fi
  \write\glswrite{(define-location-class
    \string"Roman-page-numbers\string"^^J\space\space\space
    (\string"roman-numbers-uppercase\string")
    :min-range-length \@glsminrange)}}%
  \write\glswrite{(define-location-class
    \string"arabic-page-numbers\string"^^J\space\space\space
    (\string"arabic-numbers\string")
    :min-range-length \@glsminrange)}}%
  \write\glswrite{(define-location-class
    \string"alpha-page-numbers\string"^^J\space\space\space
    (\string"alpha\string")
    :min-range-length \@glsminrange)}}%

```

```

\write\glswrite{(define-location-class
  \string"Alpha-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string")
    :min-range-length \@glsmminrange)}%
\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 \@glsmminrange)}%
\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 \@glsmminrange)}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\glsseeformat\string"
  :close \string"{}\string")}%
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
  :open \string"\string
  \glossarysection[\string\glossarytoctitle]{\string
  \glossarytitle}\string\glossarypreamble\string~n\string\begin
  {theglossary}\string\glossaryheader\string~n\string" ^^J\space
  \space\space:close \string"\expandafter\@gobble
  \string\%\string~n\string
  \end{theglossary}\string\glossarypostamble
  \string~n\string" ^^J\space\space\space
  :tree)}}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\string~n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \string~n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list

```

```

:sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
:sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\string~\n\string")}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
:open-head \string"\string\glsgroupheading
\glsopenbrace\string"^^J\space\space\space
:close-head \string"\glsclosebrace\string")}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%
\noist}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}
\def\writeist{\relax
\openout\glswrite=\istfilename
\write\glswrite{\expandafter\@gobble\string\% makeindex style file
created by the glossaries package}
\write\glswrite{\expandafter\@gobble\string\% for document
'\jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\@gls@actualchar'}
\write\glswrite{encap '\@gls@encapchar'}
\write\glswrite{level '\@gls@levelchar'}
\write\glswrite{quote '\@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
\glossarytoctitle]{\string\glossarytitle}\string
\glossarypreamble\string\n\string\begin{theglossary}\string
\glossaryheader\string\n\string"}

```

```

\write\glswrite{postamble \string"\string%\string\n\string
  \end{theglossary}\string\glossarypostamble\string\n
  \string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
  \string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}
\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
  \string"\string\relax \string\glsresetentrylist\string\n
  \string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
  \string"\string\relax \string\glsresetentrylist\string\n
  \string"}
\write\glswrite{delim_0 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_2 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_t \string"\string\}\string}\string"}
\write\glswrite{delim_n \string"\string\delimN \string"}
\write\glswrite{delim_r \string"\string\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
  \string"\string\}\string\relax
  \string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"glscpositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\noist
}
\fi
\renewcommand*{\noist}{\let\writeist\relax}

```

8.22 Rollback v4.49 (glossaries-compatible-307-2021-11-01.sty)

Version 4.49 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-307}[2021/11/01 v4.49 (NLCT)]
\newcommand{\compatglossarystyle}[2]{%
  \ifcsundef{@glscompstyle@#1}%
  {%
    \csdef{@glscompstyle@#1}{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary compatibility style ‘#1’ is already defined}{}%
  }%
}
\compatglossarystyle{inline}{%
  \renewcommand{\glossaryentryfield}[5]{%
    \glsinlinedopostchild
    \gls@inlinesep
    \def\glo@desc{##3}%
    \def\@no@post@desc{\nopostdesc}%
    \glsentryitem{##1}\glsinlinenameformat{##1}{##2}%
    \ifx\glo@desc\@no@post@desc
      \glsinlineemptydescformat{##4}{##5}%
    \else
      \ifstrempy{##3}%
        {\glsinlineemptydescformat{##4}{##5}}%
        {\glsinlinedescformat{##3}{##4}{##5}}%
      \fi
      \ifglshaschildren{##1}%
      {%
        \glsresetsubentrycounter
        \glsinlineparentchildseparator
        \def\gls@inlinesubsep{}%
        \def\gls@inlinepostchild{\glsinlinepostchild}%
      }%
      {}%
    \def\gls@inlinesep{\glsinlineseparator}%
  }%
  \renewcommand{\glossarysubentryfield}[6]{%
    \gls@inlinesubsep%
    \glsinlinesubnameformat{##2}{##3}%
    \glssubentryitem{##2}\glsinlinesubdescformat{##4}{##5}{##6}%
    \def\gls@inlinesubsep{\glsinlinesubseparator}%
  }%
}
\compatglossarystyle{list}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \item[\glsentryitem{##1}\glstarget{##1}{##2}]
```

```

        ##3\glspostdescription\space ##5}%
\renewcommand*{\glossarysubentryfield}[6]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6.}%
}
\compatglossarystyle{listgroup}{%
    \csuse{@glscmpstyle@list}%
}%
\compatglossarystyle{listhypergroup}{%
    \csuse{@glscmpstyle@list}%
}%
\compatglossarystyle{altlist}{%
    \renewcommand*{\glossaryentryfield}[5]{%
        \item[\glssentryitem{##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{@glscmpstyle@altlist}%
    }%
    \compatglossarystyle{altlisthypergroup}{%
        \csuse{@glscmpstyle@altlist}%
    }%
    \compatglossarystyle{listdotted}{%
        \renewcommand*{\glossaryentryfield}[5]{%
            \item[\makebox[\glslstdottedwidth][l]{%
                \glssentryitem{##1}\glstarget{##1}{##2}%
                \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##3}%
            \renewcommand*{\glossarysubentryfield}[6]{%
                \item[\makebox[\glslstdottedwidth][l]{%
                    \glssubentryitem{##2}%
                    \glstarget{##2}{##3}%
                    \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##4}%
                }%
            \compatglossarystyle{sublistdotted}{%
                \csuse{@glscmpstyle@listdotted}%
            }%
            \renewcommand*{\glossaryentryfield}[5]{%
                \item[\glssentryitem{##1}\glstarget{##1}{##2}]{%
                    & ##3\glspostdescription\space ##5\\}%
                \renewcommand*{\glossarysubentryfield}[6]{%
                    &
                    \glssubentryitem{##2}%
                    \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
                }%
            }%
        }%
    \compatglossarystyle{long}{%
        \renewcommand*{\glossaryentryfield}[5]{%
            \glssentryitem{##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]{%
    \glentryitem{##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]{%
    \glentryitem{##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]{%
    \glstryitem{##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]{%
    \glstryitem{##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]{%
    \glstryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%

```

```

\compatglossarystyle{altlongragged4colheader}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colborder}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colheaderborder}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{index}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item\glstarget{##1}{\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
    \glstarget{##1}{\textbf{\glstarget{##1}{##2}}}%
    \ifx\relax##4\relax
      \else
        \space(##4)%
      \fi
    \glstarget{##1}{\textbf{\glstarget{##1}{##2}}}%
    \ifx\relax##5\relax
      \else
        \space(##5)%
      \fi
    \space##4\glspostdescription\space ##6}%
}%

```

```

\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\textbf{\glstarget{##2}{##3}}%
\ifx\relax##5\relax
\else
\space(##5)%
\fi
\space##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treegroup}{%
\csuse{@glscmpstyle@tree}%
}%
\compatglossarystyle{treehypergroup}{%
\csuse{@glscmpstyle@tree}%
}%
\compatglossarystyle{treenoname}{%
\renewcommand{\glossaryentryfield}[5]{%
\hangindent0pt\relax
\parindent0pt\relax
\glssubentryitem{##1}\textbf{\glstarget{##1}{##2}}%
\ifx\relax##4\relax
\else
\space(##4)%
\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\glstarget{##2}{\strut}%
##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treenonamegroup}{%
\csuse{@glscmpstyle@treenoname}%
}%
\compatglossarystyle{treenonamehypergroup}{%
\csuse{@glscmpstyle@treenoname}%
}%
\compatglossarystyle{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]{%
\glstentryitem{##1}\textbf{\glstarget{##1}{##2}}}%
\ifx\relax##4\relax
\else
(##4)\space
\fi
##3\glspostdescription\space ##5\par
\def\@gls@prevlevel{0}%
}%
\renewcommand{\glossarysubentryfield}[6]{%
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\ifnum\@gls@prevlevel=##1\relax
\else
\@ifundefined{\@glswidestname\romannumeral##1}{%
\settowidth{\@gls@tmplen}{\textbf{\@glswidestname\space}}{%
\settowidth{\@gls@tmplen}{\textbf{%
\csname \@glswidestname\romannumeral##1\endcsname\space}}}%
\ifnum\@gls@prevlevel<##1\relax
\setlength\glstreeindent\@gls@tmplen
\addtolength\glstreeindent\parindent
\parindent\glstreeindent
\else
\@ifundefined{\@glswidestname\romannumeral\@gls@prevlevel}{%
\settowidth{\glstreeindent}{\textbf{%
\@glswidestname\space}}}%
\settowidth{\glstreeindent}{\textbf{%
\csname \@glswidestname\romannumeral\@gls@prevlevel
\endcsname\space}}}%
\addtolength\parindent{-\glstreeindent}%
\setlength\glstreeindent\parindent
\fi
\fi
\hangindent\glstreeindent
\makebox[0pt][r]{\makebox[\@gls@tmplen][l]{%
\textbf{\glstarget{##2}{##3}}}%
\ifx##5\relax\relax
\else
(##5)\space
\fi
##4\glspostdescription\space ##6\par
\def\@gls@prevlevel{##1}%
}%
}%
\compatglossarystyle{almtreeegroup}{%

```

```

\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{almtreehypergroup}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolindex}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexgroup}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexhypergroup}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcoltree}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreename}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenamegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenamehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolalmtree}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolalmtreegroup}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolalmtreehypergroup}{%
\csuse{@glscompstyle@almtree}%
}%

\compatglossarystyle{superragged}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glstentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4\glspostdescription\space ##6%

```

```

\tabularnewline}%
}%
\compatglossarystyle{superraggedborder}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheader}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheaderborder}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superragged3col}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{superragged3colborder}{%
\csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheader}{%
\csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheaderborder}{%
\csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{altsuperragged4col}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%
\compatglossarystyle{altsuperragged4colheader}{%
\csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colborder}{%
\csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colheaderborder}{%
\csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{super}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
\renewcommand*{\glossarysubentryfield}[6]{%
&

```

```

\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
}%
\compatglossarystyle{superborder}{%
\csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{superheader}{%
\csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{superheaderborder}{%
\csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{super3col}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4 & ##6\\}%
}%
\compatglossarystyle{super3colborder}{%
\csuse{@glscmpstyle@super3col}%
}%
\compatglossarystyle{super3colheader}{%
\csuse{@glscmpstyle@super3col}%
}%
\compatglossarystyle{super3colheaderborder}{%
\csuse{@glscmpstyle@super3col}%
}%
\compatglossarystyle{super4col}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
}%
\compatglossarystyle{super4colheader}{%
\csuse{@glscmpstyle@super4col}%
}%
\compatglossarystyle{super4colborder}{%
\csuse{@glscmpstyle@super4col}%
}%
\compatglossarystyle{super4colheaderborder}{%
\csuse{@glscmpstyle@super4col}%
}%
\compatglossarystyle{altsuper4col}{%
\csuse{@glscmpstyle@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]{%
\ifcsemtty{glo@\glsdetoklabel{#1}@prefix}%
{#3}%
{#2}%
}
\newcommand*{\ifglshasprefixplural}[3]{%
\ifcsemtty{glo@\glsdetoklabel{#1}@prefixplural}%
{#3}%

```

```

    {#2}%
}
\newcommand*{\ifglshasprefixfirst}[3]{%
  \ifcempty{glo@glsdetoklabel{#1}@prefixfirst}%
    {#3}%
    {#2}%
}
\newcommand*{\ifglshasprefixfirstplural}[3]{%
  \ifcempty{glo@glsdetoklabel{#1}@prefixfirstplural}%
    {#3}%
    {#2}%
}
\providecommand{\@glsprefix@record@hook}[2]{%
  \ifdef\@glsxtr@record
    {\@glsxtr@record{#1}{#2}{glslink}}%
    {}%
}
\newcommand{\glsprefixsep}{}
\newrobustcmd{\pgls}{\@glshyp@opt\@pgls}
\newcommand*{\@pgls}[2][{}]{%
  \new@ifnextchar[%
    {\@pgls@{#1}{#2}}%
    {\@pgls@{#1}{#2}[]}%
}
\def\@pgls@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdofexists{#2}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefix{#2}{\glsentryprefix{#2}\glsprefixsep}{}%
    }%
    {%
      \ifglshasprefixfirst{#2}{\glsentryprefixfirst{#2}\glsprefixsep}{}%
    }%
    \@glsp@{#1}{#2}[#3]%
  }%
}
\newrobustcmd{\pglsp1}{\@glshyp@opt\@pglsp1}
\newcommand*{\@pglsp1}[2][{}]{%
  \new@ifnextchar[%
    {\@pglsp1@{#1}{#2}}%
    {\@pglsp1@{#1}{#2}[]}%
}
\def\@pglsp1@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdofexists{#2}%
  {%
    \ifglused{#2}%
    {%

```

```

        \ifglshasprefixplural{#2}{\glstryentryprefixplural{#2}\glsprefixsep}{}%
    }%
    {%
        \ifglshasprefixfirstplural{#2}%
            {\glstryentryprefixfirstplural{#2}\glsprefixsep}{}%
        }%
        \@glsp1@{#1}{#2}[#3]%
    }%
}
\newrobustcmd{\Pgls}{\@glshyp@opt\Pgls}
\newcommand*{\@Pgls}[2][{}]{%
    \new@ifnextchar[%
        {\@Pgls@{#1}{#2}}%
        {\@Pgls@{#1}{#2}[]}%
    }
\def\@Pgls@#1#2[#3]{%
    \@glsprefix@record@hook{#1}{#2}%
    \glstrydoifexists{#2}%
    {%
        \ifglssused{#2}%
        {%
            \ifglshasprefix{#2}%
            {%
                \glstryentryprefix{#2}%
                \glsprefixsep
                \@glss@{#1}{#2}[#3]%
            }%
            {\@glss@{#1}{#2}[#3]}%
        }%
        {%
            \ifglshasprefixfirst{#2}%
            {%
                \glstryentryprefixfirst{#2}%
                \glsprefixsep
                \@glss@{#1}{#2}[#3]%
            }%
            {\@glss@{#1}{#2}[#3]}%
        }%
    }%
}
\newrobustcmd{\Pglspl}{\@glshyp@opt\Pglspl}
\newcommand*{\@Pglspl}[2][{}]{%
    \new@ifnextchar[%
        {\@Pglspl@{#1}{#2}}%
        {\@Pglspl@{#1}{#2}[]}%
    }
\def\@Pglspl@#1#2[#3]{%
    \@glsprefix@record@hook{#1}{#2}%
    \glstrydoifexists{#2}%
    {%

```

```

\ifglused{#2}%
{%
  \ifglshasprefixplural{#2}%
  {%
    \Glsentryprefixplural{#2}%
    \glsprefixsep
    \@glsp1@{#1}{#2}[#3]%
  }%
  {\@Glspl@{#1}{#2}[#3]}%
}%
{%
  \ifglshasprefixfirstplural{#2}%
  {%
    \Glsentryprefixfirstplural{#2}%
    \glsprefixsep
    \@glsp1@{#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}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefix{#2}%
      {\mfirstucMakeUppercase{\glentryprefix{#2}\glsprefixsep}}}%
    }%
    {%
      \ifglshasprefixfirst{#2}%
      {\mfirstucMakeUppercase{\glentryprefixfirst{#2}\glsprefixsep}}}%
    }%
    \@GLS@{#1}{#2}[#3]%
  }%
}
\newrobustcmd{\PGLSp1}{\@gls@hyp@opt\PGLSp1}
\newcommand*{\@PGLSp1}[2][{}]{%
  \new@ifnextchar[%
    {\@PGLSp1@{#1}{#2}}%
    {\@PGLSp1@{#1}{#2}[]}%
}
\def\@PGLSp1@#1#2[#3]{%

```

```

\@glsprefix@record@hook{#1}{#2}%
\glsdoifexists{#2}%
{%
  \ifglsused{#2}%
  {%
    \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@hypergroup@#1\endcsname
  \@for\@gls@elem:=\@gls@list\do{%
    \ifthenelse{\equal{\@gls@elem}{#2}}{\@endfortrue}{}%
  }
  \if@endfor
  \else
    \GlossariesWarningNoLine{Navigation panel
      for glossary type ‘#1’^^Jmissing group ‘#2’}%
    \gdef\gls@hypergroup@rerun{%
      \GlossariesWarningNoLine{Navigation panel
        has changed. Rerun LaTeX}}%
  \fi
}
\let\gls@hypergroup@rerun\relax
\AtEndDocument{\gls@hypergroup@rerun}
\newcommand*{\@gls@hypergroup}[2]{%
  \@ifundefined{\@gls@hypergroup@list@#1}{%
    \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{#2}%
  }{%

```

```

\expandafter\let\expandafter\@gls@tmp
\csname @gls@hypergroupplist@#1\endcsname
\expandafter\xdef\csname @gls@hypergroupplist@#1\endcsname{%
\@gls@tmp,#2}%
}%
}
\newcommand*{\glsnavigation}{%
\def\@gls@between{}%
\ifcsundef\@gls@hypergroupplist@\@glo@type}%
{%
\def\@gls@list{}%
}%
{%
\expandafter\let\expandafter\@gls@list
\csname @gls@hypergroupplist@\@glo@type\endcsname
}%
\@for\@gls@tmp:=\@gls@list\do{%
\@gls@between
\@gls@getgrouptitle{\@gls@tmp}{\@gls@grptitle}%
\glsnavhyperlink{\@gls@tmp}{\@gls@grptitle}%
\let\@gls@between\glshypernavsep
}%
}
\newcommand*{\glshypernavsep}{\space\textbar\space}
\newcommand*{\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}%
}%
\ifglshasdescsuppressed{##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}}%
  \glsesubentryitem{##2}%
  \glsinlinesubdescformat{\glossentrydesc{##2}}{\glossentrysymbol{##2}}{##3}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%
\renewcommand*{\glsgroupskip}{}%
}
\newcommand*{\glsinlinedopostchild}{%
  \gls@inlinepostchild
  \def\gls@inlinepostchild{}%
}
\newcommand*{\glsinlineseparator}{;\space}
\newcommand*{\glsinlinesubseparator}{,\space}
\newcommand*{\glsinlineparentchildseparator}{:\space}
\newcommand*{\glsinlinepostchild}{}
\newcommand*{\glspostinline}{\glspostdescription\space}
\newcommand*{\glsinlinenameformat}[2]{\glstarget{#1}{#2}}
\newcommand*{\glsinlinedescformat}[3]{\space#1}
\newcommand*{\glsinlineemptydescformat}[2]{}

```



```

\newcommand*{\glsinlinesubnameformat}[2]{\glstarget{#1}{}}
\newcommand*{\glsinlinesubdescformat}[3]{#1}

```

8.27 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
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
  \fi
}
\newglossarystyle{longborder}{%
  \setglossarystyle{long}%
  \renewenvironment{theglossary}{%
    \begin{longtable}[|lp{glsdescwidth}|]{\end{longtable}}%
    \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
  }
\newglossarystyle{longheader}{%
  \setglossarystyle{long}%
  \renewcommand*{\glossaryheader}{%
    \bfseries \entryname & \bfseries \descriptionname\tabularnewline\endhead}%
  }

```

```

\newglossarystyle{longheaderborder}{%
  \setglossarystyle{longborder}%
  \renewcommand*{\glossaryheader}{%
    \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}\{ll|p{\glsdescwidth}|p{\glspagelistwidth}|\}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long3colheader}{%
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
  }
\newglossarystyle{long3colheaderborder}{%
  \setglossarystyle{long3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
  }

```

```

}
\newglossarystyle{long4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{long4colheader}{%
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long4colborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long4colheaderborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|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{\glsgdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheader}{%
  \setglossarystyle{long4colheader}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colborder}{%
  \setglossarystyle{long4colborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|lp{\glsgdescwidth}|lp{\glspagelistwidth}|}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheaderborder}{%
  \setglossarystyle{long4colheaderborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|lp{\glsgdescwidth}|lp{\glspagelistwidth}|}%
    {\end{longtable}}%
}
}

```

8.29 Rollback v4.49 (glossary-longbooktabs-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-longbooktabs}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{booktabs}
\RequirePackage{glossary-long}[=v4.46]
\RequirePackage{glossary-longragged}[=v4.46]
\newglossarystyle{long-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname & \bfseries
    \descriptionname\tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{long3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%

```

```

        \toprule \bfseries \entryname &
        \bfseries \descriptionname &
        \bfseries \pagelistname
        \tabularnewline\midrule\endhead
        \bottomrule\endfoot}%
\ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
\else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}
\newglossarystyle{long4col-booktabs}{%
    \glspatchLToutput
    \setglossarystyle{long4col}%
    \renewcommand*{\glossaryheader}{%
        \toprule \bfseries \entryname &
        \bfseries \descriptionname &
        \bfseries \symbolname &
        \bfseries \pagelistname
        \tabularnewline\midrule\endhead
        \bottomrule\endfoot}%
    \ifglsgroupskip
        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
    \fi
}
\newglossarystyle{altlong4col-booktabs}{%
    \glspatchLToutput
    \setglossarystyle{long4col-booktabs}%
    \renewenvironment{theglossary}%
        {\begin{longtable}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
        {\end{longtable}}%
}
\newglossarystyle{longragged-booktabs}{%
    \glspatchLToutput
    \setglossarystyle{long-booktabs}%
    \renewenvironment{theglossary}%
        {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}}}%
        {\end{longtable}}%
}
\newglossarystyle{longragged3col-booktabs}{%
    \glspatchLToutput
    \setglossarystyle{long3col-booktabs}%
    \renewenvironment{theglossary}%
        {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}%
            >{\raggedright}p{\glspagelistwidth}}}%
        {\end{longtable}}%
}
\newglossarystyle{altlongragged4col-booktabs}{%

```

```

\glspatchLToutput
\setglossarystyle{altlong4col-booktabs}%
\renewenvironment{theglossary}%
  {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
    >{\raggedright}p{\glspagelistwidth}}}%
  {\end{longtable}}%
}
\newcommand*{\glsLTpenaltycheck}{%
  \ifnum\outputpenalty=-50\vskip-\normalbaselineskip\relax\fi
}
\newcommand{\glspenaltygroupskip}{%
  \noalign{\penalty-50\vskip\normalbaselineskip}}
\let\@gls@org@LT@output\LT@output
\newcommand*{\glsrestoreLToutput}{\let\LT@output\@gls@org@LT@output}
\newcommand*{\glspatchLToutput}{%
  \renewcommand*{\LT@output}{%
    \ifnum\outputpenalty <-\@Mi
      \ifnum\outputpenalty > -\LT@end@pen
        \LT@err{floats and marginpars not allowed in a longtable}\@ehc
      \else
        \setbox\z@\vbox{\unvbox\@cclv}%
        \ifdim \ht\LT@lastfoot>\ht\LT@foot
          \dimen@\pagegoal
          \advance\dimen@-\ht\LT@lastfoot
          \ifdim\dimen@<\ht\z@
            \setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%
            \@makecol
            \@outputpage
            \setbox\z@\vbox{\box\LT@head\glsLTpenaltycheck}%
          \fi
          \fi
          \global\@colroom\@colht
          \global\vsizel\@colht
          {\unvbox\z@\box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%
        \fi
      \else
        \setbox\@cclv\vbox{\unvbox\@cclv\copy\LT@foot\vss}%
        \@makecol
        \@outputpage
        \global\vsizel\@colroom
        \copy\LT@head
        \glsLTpenaltycheck
        \nobreak
      \fi
    }%
  }
}

```


8.30 Rollback v4.49 (glossary-longragged-2021-11-01.sty)

Version 4.49 preserved for rollback.

```
\ProvidesPackage{glossary-longragged}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{array}
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{longragged}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}[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
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{longragged3colborder}{%
  \setglossarystyle{longragged3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}[ll>{\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{\glsgrouptitle{##1}}\indexspace}%
}
\newglossarystyle{mcolindexhypergroup}{%
\setglossarystyle{mcolindex}%
\renewcommand*{\glossaryheader}{%
\item\glstreenavigationfmt{\glsnavigation}\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}}{\glsgrouptitle{##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{\glsgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreenonamehypergroup}{%
  \setglossarystyle{mcoltreenoname}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##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}{\glsgrouptitle{##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{mcolalittreehypergroup}{%
\setglossarystyle{mcolalittree}%
\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{mcolalittreespannav}{%
\setglossarystyle{alittree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsncols}\noindent\glstreenavigationfmt{\glsnavigation}]
\def\@gls@prevlevel{-1}%
\mbox{}\par
}%
{\par\end{multicols}}%
\renewcommand*\glsgroupheading[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}

```

8.32 Rollback v4.49 (glossary-super-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-super}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
\newlength{glsdescwidth}
\setlength{glsdescwidth}{0.6\hspace}
}{}

```

```

\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{super}{%
  \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}}%
    {\begin{supertabular}{lp{\glstdescwidth}}}%
    {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupeheading}[1]{%
  \renewcommand{\glossentry}[2]{%
    \glstentryitem{##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
  }%
  \ifglsgnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    {\begin{supertabular}{|lp{\glstdescwidth}|}%
    {\end{supertabular}}}%
}
\newglossarystyle{superheader}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries \entryname &
    \bfseries\descriptionname\tabularnewline}%
    \tabletail{}%
    {\begin{supertabular}{lp{\glstdescwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{superheaderborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline\bfseries \entryname &
    \bfseries \descriptionname\tabularnewline\hline}%
    \tabletail{\hline}
    {\begin{supertabular}{|lp{\glstdescwidth}|}%

```



```

        {\end{supertabular}}}%
    }
\newglossarystyle{super3col}{%
    \renewenvironment{theglossary}%
        {\tablehead{}\tabletail{}}%
        {\begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}}%
        {\end{supertabular}}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
        \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1} & ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2} &
        ##3\tabularnewline
    }%
    \ifglsgroupskip
        \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}{|l|l|l|l|}{%
      \end{supertabular}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
  \fi
}
\newglossarystyle{super4colheader}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}%
    {\tabletail{}}%
    {\begin{supertabular}{|l|l|l|l|}{%
      \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{\glstdescwidth}lp{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuper4colheader}{%
  \setglossarystyle{super4colheader}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    {\begin{supertabular}{lp{\glstdescwidth}lp{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuper4colborder}{%
  \setglossarystyle{super4colborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}%
      {\lllp{\glstdescwidth}\lllp{\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}%
      {\lllp{\glstdescwidth}\lllp{\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{glstdescwidth}{%
  \newlength{glstdescwidth}
  \setlength{glstdescwidth}{0.6\hspace}
}{%
}

```

```

\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{superragged}{%
  \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}}%
    {\begin{supertabular}{l>{\raggedright}p{\glstdescwidth}}}%
    {\end{supertabular}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupeheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2%
    \tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3%
    \tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superraggedborder}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    {\begin{supertabular}{|l|>{\raggedright}p{\glstdescwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{superraggedheader}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries \entryname & \bfseries \descriptionname
    \tabularnewline}%
    \tabletail{}}%
    {\begin{supertabular}{l>{\raggedright}p{\glstdescwidth}}}%
    {\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
}%
\ifglsgroupskip
\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{\glsgdescwidth}|%
        >{\raggedright}p{\glspagelistwidth}|}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuperragged4col}{%
    \renewenvironment{theglossary}%
    {\tablehead{} \tabletail{}}%
    \begin{supertabular}{|l>{\raggedright}p{\glsgdescwidth}l%
        >{\raggedright}p{\glspagelistwidth}|}%
    {\end{supertabular}}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
        \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1} &
        \glossentrysymbol{##1} & ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2} &
        \glossentrysymbol{##2} & ##3\tabularnewline
    }%
    \ifglsgroupskip
        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
    \fi
}
\newglossarystyle{altsuperragged4colheader}{%
    \setglossarystyle{altsuperragged4col}%
    \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
        \bfseries\symbolname &
        \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    \begin{supertabular}{|l>{\raggedright}p{\glsgdescwidth}l%
        >{\raggedright}p{\glspagelistwidth}|}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuperragged4colborder}{%
    \setglossarystyle{altsuper4col}%
    \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}%
        {\l|>{\raggedright}p{\glsgdescwidth}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}%
      {|||>{\raggedright}p{\glsdescwidth}|||}%
      >{\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{\glsgrouptitle{##1}}%
\indexspace
}%
}
\newglossarystyle{indexhypergroup}{%
\setglossarystyle{index}%
\renewcommand*{\glossaryheader}{%
\item\glstreenavigationfmt{\glsnavigation}\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavigationtarget{##1}{\glsgrouptitle{##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{\glsgrouptitle{##1}}\par
    \indexspace}%
}
\newglossarystyle{treehypergroup}{%
  \setglossarystyle{tree}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt
    {\glshypertarget{##1}{\glsgrouptitle{##1}}}\par
    \indexspace}%
}
\newlength\glstreeindent
\setlength{\glstreeindent}{10pt}
\newglossarystyle{treenoname}{%
  \renewenvironment{theglossary}%
    {\setlength{\parindent}{0pt}%
     \setlength{\parskip}{0pt plus 0.3pt}}%
    {}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glssentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
    \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
    \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
  }%
  \renewcommand{\subglossentry}[3]{%
    \hangindent##1\glstreeindent\relax
    \parindent##1\glstreeindent\relax
    \ifnum##1=1\relax
      \glssubentryitem{##2}%
    \fi

```

```

\glstarget{##2}{\strut}%
\glossentrydesc{##2}\glspostdescription\space##3\par
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{treenonamegroup}{%
\setglossarystyle{treenoname}%
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt
{\glsgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{treenonamehypergroup}{%
\setglossarystyle{treenoname}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt
{\glslshhypertarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}
\newrobustcmd*{\glswidesttoplevelname}[1][\@glo@types]{%
\dimen@=0pt\relax
\gls@tmplen=0pt\relax
\forall glossaries[#1]{\@gls@type}%
{%
\forall glsentries[\@gls@type]{\@glo@label}%
{%
\ifglshasparent{\@glo@label}%
}%
{%
\settowidth{\dimen@}%
{\glstreenamefmt{\glstreenamename{\@glo@label}}}%
\ifdim\dimen@>\gls@tmplen
\gls@tmplen=\dimen@
\letcs{\@glswidestname}{\glo@{\glstetoklabel{\@glo@label}@name}%
\fi
}%
}%
}%
}
\newcommand*{\glswidest}[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[Opt][r]{\glstreenamebox{\glstreeindent}{%
    \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}}}%
  \ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}%
  \glossentrydesc{##1}\glspostdescription \space ##2\par
  \def\@gls@prevlevel{0}%
}%
\renewcommand{\subglossentry}[3]{%
  \ifnum##1=1\relax
  \glssubentryitem{##2}%
  \fi
  \ifnum\@gls@prevlevel=##1\relax
  \else
    \@ifundefined{@glswidestname\romannumeral##1}{%
      \settowidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}{%
      \settowidth{\gls@tmplen}{\glstreenamefmt{%
        \csname @glswidestname\romannumeral##1\endcsname\space}}}%
      \ifnum\@gls@prevlevel<##1\relax
        \setlength\glstreeindent\gls@tmplen
        \addtolength\glstreeindent\parindent
        \parindent\glstreeindent
      \else
        \@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
          \settowidth{\glstreeindent}{\glstreenamefmt{%
            \@glswidestname\space}}{%
            \settowidth{\glstreeindent}{\glstreenamefmt{%
              \csname @glswidestname\romannumeral\@gls@prevlevel
                \endcsname\space}}}%
            \addtolength\parindent{-\glstreeindent}%
            \setlength\glstreeindent\parindent
          \fi
          \fi
          \hangindent\glstreeindent
          \makebox[Opt][r]{\glstreenamebox{\gls@tmplen}{%
            \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
          \ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
          \glossentrydesc{##2}\glspostdescription\space ##3\par
          \def\@gls@prevlevel{##1}%

```

```

}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{almtreegroup}{%
\setglossarystyle{almtree}%
\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
\par\indexspace}%
}
\newglossarystyle{almtreehypergroup}{%
\setglossarystyle{almtree}%
\renewcommand*{\glossaryheader}{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}}

```

Change History

1.01 – 2007-05-17	
General: Added range facility in format key	124
\writeist: Added spaces after \delimN and \delimR in ist file	177
1.04 – 2007-08-03	
General: Added \glstextformat	111
1.05 – 2007-08-10	
\glossarysection: added	
\mkboth to	
\glossarysection	49
\gls@defglossaryentry:	
Changed the default value of the sort key to just the value of the name key	93
1.07 – 2007-09-13	
\@gls@link: fixed bug caused by \theglsentrycounter setting the page number too soon	121
\glsadd: fixed bug caused by \theglsentrycounter setting the page number too soon	174
1.08 – 2007-10-13	
General: Added babel support	43
altlistgroup: changed altlistgroup style to use \glsgetgrouptitle	275
listgroup: changed listgroup style to use \glsgetgrouptitle	274
1.1 – 2008-02-22	
General: amsgen now loaded (\new@ifnextchar needed)	4
\@glossarysection: numbered sections and auto label added	50
\@gls@tmpb: changed \toksdef to \newtoks	126
\@gls@toc: numberline added	52
\@p@glossarysection: numbered sections and auto label added	51
numberedsection:	
numberedsection package option added	8
numberline: numberline option added	8
\setglossarysection: new	50
translate: translate option added	28
1.12 – 2008-03-08	
General: added check for \hypertarget separate to \hyperlink (memoir defines \hyperlink but not \hypertarget)	133
\@GLSpl: now uses \glentrydescplural and \glentrysymbolplural instead of \glentrydesc and \glentrysymbol	138
\@GLspl0: now uses \glentrydescplural and \glentrysymbolplural instead of \glentrydesc and \glentrysymbol	137
\@GLspl0: now uses \glentrydescplural and \glentrysymbolplural instead of \glentrydesc and \glentrysymbol	136
descriptionplural: new	73
\gls@defglossaryentry:	
Changed default first plural to be first key with s appended (was text key with s appended)	93
descriptionplural support added	93
symbolplural support added	93
\Glsentrydescplural: New	166
\glentrydescplural: New	166
\Glsentrysymbolplural: New	167
\glentrysymbolplural: New	167
symbolplural: new	75
1.13 – 2008-05-10	
General: fixed bug that ignored 3rd parameter	140–148
\ACRfullpl: new	242
\Acrfullpl: new	242
\acrfullpl: new	242
\acrpluralsuffix: New	239
\gls@defglossaryentry:	
Changed default first value	93

Changed default firstplural value	93	checking if footnote option has been used	137
Removed restriction on only using <code>\newglossaryentry</code> in the preamble	98	<code>\@gls@</code> : Test glossary type is <code>\acronymtype</code> in addition to checking if footnote option has been used	134
<code>\newacronym</code> : Removed restriction on only using <code>\newacronym</code> in the preamble	238	<code>\@glsdisp</code> : Test glossary type is <code>\acronymtype</code> in addition to checking if footnote option has been used	139
1.14 – 2008-06-17		<code>\@glspl@</code> : Test glossary type is <code>\acronymtype</code> in addition to checking if footnote option has been used	136
General: added nonumberlist key to <code>\printglossary</code>	226	<code>\@gls@</code> : raised the hypertarget so the target text doesn't scroll off the top of the page	133
added numberedsection key to <code>\printglossary</code>	224	<code>\gls@defglossaryentry</code> : Changed def to let	93
<code>\@gls@hypergroup</code> : new	269	1.17 – 2008-12-26	
<code>\firstacronymfont</code> : new	243	General: added xindy support	30
<code>\glsautoprefix</code> : new	8	<code>\@do@esc@wrglossary</code> : new	203
<code>\glsnavhyperlink</code> : changed <code>\edef</code> to <code>\protected@edef</code>	268	<code>\@do@seeglossary</code> : new	207
<code>\glsnavhypertarget</code> : added write to aux file	268	<code>\@glo@storeentry</code> : new	100
<code>\glsnavigation</code> : changed to only use labels for groups that are present	269	<code>\@gls@glossary</code> : changed definition to use <code>\index</code> instead of <code>\@index</code>	199
1.15 – 2008-08-15		<code>\@glsdefaultplural</code> : new	78
<code>\@gls@link</code> : added <code>\glslabel</code>	121	<code>\@glsdefaultsort</code> : new	79
<code>\gls@defglossaryentry</code> : check for <code>\@glo@first</code> in description	97	<code>\@gls@hypernumber</code> : new	235
check for <code>\@glo@text</code> in symbol	97	<code>\@gls@noname</code> : new	78
<code>\gls@hypergroup</code> : new	269	<code>\@gls@nonextpages</code> : new	226
<code>\glsnavhypertarget</code> : added check if rerun required	268	<code>\gls@defglossaryentry</code> : added nonumberlist key	94
<code>\glssettoctitle</code> : new	42	added parent key	93
<code>\printglossary</code> : changed the way the TOC title is set	210	added see key	93
1.16 – 2008-08-27		Stored main part of entry format when entry is defined	98
<code>\@GLS@</code> : Test glossary type is <code>\acronymtype</code> in addition to checking if footnote option has been used	136	<code>\gls@suffixF</code> : new	47
<code>\@GLSpl</code> : Test glossary type is <code>\acronymtype</code> in addition to checking if footnote option has been used	138	<code>\gls@suffixFF</code> : new	47
<code>\@Gls@</code> : Test glossary type is <code>\acronymtype</code> in addition to checking if footnote option has been used	135	<code>\gls@wrglossary</code> : modified to allow for xindy support	200
<code>\@Glspl</code> : Test glossary type is <code>\acronymtype</code> in addition to		<code>\gls@hyperlink</code> : new	173
		<code>\gls@hypernumber</code> : modified to allow material to be attached to location	235
		<code>\glsnavhyperlink</code> : replaced <code>\hyperlink</code> to <code>\@glslink</code>	268

\glsnavhypertarget: replaced	\@printglossary: changed \xdef
\hypertarget to	to \protected\xdef 213
\@gls@target 268	\gls@defglossaryentry: changed
\glssee: new 208	\xdef to
\glsseeformat: new 208	\protected\xdef 95, 98
\glsSetSuffixF: new 47	\glsaddprotectedpagefmt:
\glsSetSuffixFF: new 47	changed \eappto to
\ifglxindy: new 30	\protected@eappto 201
\istfilename: added xindy	2.01 – 2009 May 30
support 46	\@gls@link: moved
\newglossarystyle: made	\@do@wrglossary before term
\newglossarystyle long . . 234	is displayed to prevent
nonumberlist: new 76	unwanted whatsit 122
\nopostdesc: new 45	\forallglossaries: replaced
order: order package option	\ifthenelse with \ifx 61
added 30	\forallglsentries: replaced
parent: new 76	\ifthenelse with \ifx 62
\printglossary: added check to	\glsdefmain: new 17
determine if \printglossary	\glsdescwidth: changed
is already defined 210	\linewidth to \hsize . 277, 300
added print language to aux	\glslistdottedwidth: changed
file 210	\linewidth to \hsize 276
see: new 75	\glspagelistwidth: changed
\writeist: added xindy support 177	\linewidth to \hsize . 277, 300
1.18 – 2009-01-14	nomain: added nomain package
\@gls@loadlist: new 11	option 18
\@gls@loadlong: new 10	\writeist: removed item_02 -
\@gls@loadsuper: new 10	no such makeindex key . . . 182
\@gls@loadtree: new 11	2.02 – 2007-07-13
\gls@defglossaryentry:	\@printglossary: suppressed
Changed default value of sort	warning globally rather than
to \glsdefaultsort 93	locally 213
moved sort sanitization to	2.02 – 2009-07-13
\newglossaryentry 97	\glossarysection: changed
\glstarget: new 229	\@mkboth to \glossarymark 49
nolist: new 11	\gls@glossarymark: New 49
nolong: new 10	2.03 – 2009-09-23
nostyles: new 11	\@GLS@: Added check for
nosuper: new 11	hyperfirst 136
notree: new 11	\@GLSpl: Added check for
\oldacronym: new 238	hyperfirst 138
sort: moved sanitization to	\@GLS@: Added check for
\newglossaryentry 74	hyperfirst 135
1.19 – 2009-03-02	\@GLspl@: Added check for
\gls@clearpage: new 52	hyperfirst 137
\glsdisp: new 138	\@gls@: Added check for
1.46 – 2021-09-20	hyperfirst 134
General: changed \edef to	\@gls@link: new 120
\protected@edef 368	\@gls@link: added \leavevmode 121

Moved entry existence check to avoid duplicate code	121	\glsadd: fixed bug that ignored counter	174
\@glsdisp: Added check for hyperfirst	139	\Glsentryuseri: new	169
\@glspl@: Added check for hyperfirst	136	\glsentryuseri: new	169
\gls glossarymark: Added check to see if it's already defined .	49	\Glsentryuserii: new	169
hyperfirst: new	29	\glsentryuserii: new	169
2.04 – 2009-11-10		\Glsentryuseriii: new	169
\@GLS@: Changed test to check if glossary type has been identified as a list of acronyms	136	\glsentryuseriii: new	169
\@GLSpl: Changed test to check if glossary type has been identified as a list of acronyms	138	\Glsentryuseriv: new	170
\@Gls@: Changed test to check if glossary type has been identified as a list of acronyms	135	\glsentryuseriv: new	170
\@Glspl@: Changed test to check if glossary type has been identified as a list of acronyms	137	\Glsentryuseriv: new	170
\@glossaryentryfield: new . .	99	\Glsentryuserv: new	170
\@glossarysubentryfield: new	99	\glsentryuserv: new	170
\@gls@: Changed test to check if glossary type has been identified as a list of acronyms	134	\Glsentryuservi: new	170
\@glsacronymlists: new	19	\glsentryuservi: new	170
\@glsdisp: Changed test to check if glossary type has been identified as a list of acronyms	139	\ns@newglossary: added check to determine if \gls@<type>@display and \gls@<type>@displayfirst have been defined.	71
\@glspl@: Changed test to check if glossary type has been identified as a list of acronyms	136	\SetAcronymLists: new	20
\@newglossaryentryposthook: new	99	\SetDefaultAcronymDisplayStyle: new	258
\@newglossaryentryprehook: new	99	\SetDefaultAcronymStyle: new	258
acronymlists: new	20	\SetDescriptionAcronymDisplayStyle: new	259
\DeclareAcronymList: new . . .	19	\SetDescriptionDUAAcronymDisplayStyle: new	259
\DefineAcronymSynonyms: new .	256	\SetDescriptionFootnoteAcronymDisplayStyle: new	258
\gls@defglossaryentry: added user1-6 keys	94	\SetDUADisplayStyle: new . . .	260
		\SetFootnoteAcronymDisplayStyle: new	259
		\SetSmallAcronymDisplayStyle: new	260
		2.05 – 2010-02-06	
		\@glsdisp: Added closing brace. Patch provided by Sergiu Dotenco	139
		Removed spurious brace. Patch provided by Sergiu Dotenco	139
		\writeist: Added \string before opening and closing braces. Patch provided by Sergiu Dotenco	182
		2.06 – 2010-06-14	
		\altnewglossary: new	71
		\CustomAcronymFields: new . .	261
		\CustomNewAcronymDef: new . .	261
		\SetCustomDisplayStyle: new .	261
		\SetCustomStyle: new	261

2.07 – 2010-07-10		\@xdyattributelist: new	52
General: glsadd format key stored		\acrfootnote: new	258
in \@glsnumberformat (was		\ACRfull: added starred version	241
mistakenly stored in		\Acrfull: added starred version	241
\@glo@format)	174	\acrfull: added starred version	241
3.0 – 2010-07-12		\ACRfullpl: added starred	
\@makeglossary: Added check for		version	242
savewrites	187	\Acrfullpl: added starred	
\@gls@wrglossary: modified to		version	242
take into account savewrites	200	\acrfullpl: added starred	
3.0 – 2010/03/31		version	242
\@set@glo@numformat: added 4th		\acrlinkfootnote: new	258
argument	124	\acrnolinkfootnote: new	258
3.0 – 2011-04-02		compatible-2.07:	
General: added prefix to		compatible-2.07 option added	36
hyperlink	236	counter: replaced \@ifundefined	
etoolbox now loaded	4	with \ifcsundef	75
replaced \@ifundefined with		counterwithin: new	12
\ifcsundef	41, 44, 117, 224	entrycounter: new	12
\@do@esc@wrglossary: added		glossaryentry: new	12
check for hyper location		\glossarysection: replaced	
prefix	205	\@ifundefined with	
modified to use new format .	203	\ifcsundef	49
\@glossarysec: replaced		\glossarystyle: replaced	
\@ifundefined with		\@ifundefined with	
\ifcsundef	8	\ifcsundef	233
\@do@seeglossary: Sanitize and		glossarysubentry: new	13
escape cross-referencing		\gls@codepage: replaced	
information	207	\@ifundefined with	
\@gls@counterwithin: new . . .	13	\ifcsundef	31
\@gls@ifinlist: new	53	\gls@defglossaryentry: added	
\@gls@link: added		\@gls@defsort	97
\@gls@saveentrycounter .	122	added short and long keys .	94
added \@gls@setsort	122	replaced \@ifundefined with	
\@gls@saveentrycounter: new .	122	\ifcsundef	94
\@gls@setupsort@def: new . . .	15	\gls@doclearpage: replaced	
\@gls@setupsort@standard: new	14	\@ifundefined with	
\@gls@setupsort@use: new . . .	15	\ifcsundef	51
\@gls@xdy@locationlist: new .	56	\glsadd: added	
\@glslink: replaced		\@gls@saveentrycounter .	174
\@ifundefined with		\GlsAddXdyCounters: new	53
\ifcsundef	132	\glentrycounterlabel: new .	228
\@glsnextpages: new	226	\glentryitem: new	228
\@print@glossary: replaced		\Glentrylong: new	171
\@ifundefined with		\glentrylong: new	171
\ifcsundef	213	\Glentrylongpl: new	171
\@printglossary: added		\glentrylongpl: new	171
\currentglossary	212	\Glentryshort: new	171
added \glsnextpages	212	\glentryshort: new	170
make toctitle default to title	212	\Glentryshortpl: new	171

<code>\glstryshorttpl</code> : new	171	<code>\oldacronym</code> : replaced	
<code>\glsgtgroupitl</code> : replaced		<code>\@ifundefined</code> with	
<code>\ifcsundef</code>	232	<code>\ifcsundef</code>	238
<code>\glsglossarymark</code> : replaced		<code>\printglossary</code> : replaced	
<code>\@ifundefined</code> with		<code>\@ifundefined</code> with	
<code>\ifcsundef</code>	49	<code>\ifcsundef</code>	210
<code>\glshyperlink</code> : changed default		<code>savewrites</code> : new	36
from <code>\glstryname</code> to		<code>see</code> : added	
<code>\glstrytext</code>	173	<code>\@glo@seeautonumberlist</code>	75
<code>\glshypernumber</code> : replaced		<code>seeautonumberlist</code> : new	10
<code>\@ifundefined</code> with		<code>\SetDescriptionFootnoteAcronymDisplayStyle</code> :	
<code>\ifcsundef</code>	235	expanded options link	
<code>\glsnrformat</code> : replaced		options	258
<code>\@ifundefined</code> with		<code>\setentrycounter</code> : added	
<code>\ifcsundef</code>	47	optional argument	233
<code>\glstryentry</code> : new	227	<code>short</code> : new	78
<code>\glstryresetentrycounter</code> :		<code>shortplural</code> : new	78
new	227	<code>\showacronymlists</code> : new	266
<code>\glstryseeitem</code> : hyperlink uses		<code>\showglocounter</code> : new	263
<code>\glstryseeitemformat</code> instead of		<code>\showglodesc</code> : new	264
<code>\glstryname</code>	209	<code>\showglodescplural</code> : new	264
<code>\glstryseeitemformat</code> : new	209	<code>\showglofirst</code> : new	262
<code>\glssortnumberfmt</code> : new	14	<code>\showglofirstpl</code> : new	262
<code>\glstrystepentry</code> : new	227	<code>\showgloflag</code> : new	265
<code>\glstrystepsubentry</code> : new	227	<code>\showgloindex</code> : new	265
<code>\glstrysubentrycounterlabel</code> :		<code>\showglolevel</code> : new	262
new	228	<code>\showgloname</code> : new	264
<code>\glstrysubentryitem</code> : new	228	<code>\showgloparent</code> : new	261
<code>\ifglossaryexists</code> : replaced		<code>\showgloplural</code> : new	262
<code>\@ifundefined</code> with		<code>\showglosort</code> : new	264
<code>\ifcsundef</code>	62	<code>\showglossaries</code> : new	266
<code>\ifglstryentryexists</code> : replaced		<code>\showglossarycounter</code> : new	267
<code>\@ifundefined</code> with		<code>\showglossaryentries</code> : new	267
<code>\ifcsundef</code>	63	<code>\showglossaryin</code> : new	266
<code>\istfile</code> : deprecated	197	<code>\showglossaryout</code> : new	266
<code>long</code> : new	78	<code>\showglossarytitle</code> : new	266
<code>longplural</code> : new	78	<code>\showglosymbol</code> : new	264
<code>\newglossaryentry</code> : replaced		<code>\showglosymbolplural</code> : new	265
<code>\DeclareRobustCommand</code> with		<code>\showgloxt</code> : new	262
<code>\newrobustcmd</code>	81	<code>\showgloxttype</code> : new	262
<code>\newglossarystyle</code> : replaced		<code>\showglouserii</code> : new	263
<code>\@ifundefined</code> with		<code>\showglouseriii</code> : new	263
<code>\ifcsundef</code>	234	<code>\showglouseriv</code> : new	263
<code>nonumberlist</code> : now boolean	76	<code>\showglouserv</code> : new	263
<code>\ns@newglossary</code> : added		<code>\showglouservi</code> : new	264
<code>\@gls@defsortcount</code>	71	<code>sort</code> : new	13
replaced <code>\@ifundefined</code> with		<code>subentrycounter</code> : new	13
<code>\ifcsundef</code>	71		

theglossary: replaced	\glsname: made robust	143
\@ifundefined with	\GLSpl: made robust	138
\ifcsundef	\Glspl: made robust	137
\writeist: added xindy-only	\glspl: made robust	136
macro definitions to glossary	\GLSplural: made robust	142
open tag	\GLSsymbol: made robust	147
modified to support new	\Glsymbol: made robust	147
format	\glsymbol: made robust	146
3.01 – 2011-04-12	\GLSsymbolplural: made robust	148
General: made robust	\Glsymbolplural: made robust	148
\@glswritefiles: added check	\glsymbolplural: made robust	147
for empty glossaries	\GLstext: made robust	140
\ACRfull: made robust	\glstext: made robust	140
\Acrfull: made robust	\GLSuseri: made robust	149
\acrfull: made robust	\Glsuseri: made robust	149
\acrfullformat: removed	\glsuseri: made robust	148
\acronymfont as it should	\GLSuserii: made robust	150
already be set in the second	\Glsuserii: made robust	150
argument.	\glsuserii: made robust	149
\ACRfullpl: made robust	\GLSuseriii: made robust	151
\Acrfullpl: made robust	\Glsuseriii: made robust	151
\acrfullpl: made robust	\glsuseriii: made robust	150
\ACRlong: made robust	\GLSuseriv: made robust	152
\Acrlong: made robust	\Glsuseriv: made robust	152
\acrlong: made robust	\glsuseriv: made robust	151
\ACRlongpl: made robust	\GLSuserv: made robust	153
\Acrlongpl: made robust	\Glsuserv: made robust	153
\acrlongpl: made robust	\glsuserv: made robust	152
\ACRshort: made robust	\GLSuservi: made robust	154
\Acrshort: made robust	\Glsuservi: made robust	154
\acrshort: made robust	\glsuservi: made robust	153
\ACRshortpl: made robust	3.02 – 2012-05-19	
\Acrshortpl: made robust	\glsnumlistlastsep: new	173
\acrshortpl: made robust	\glsnumlistsep: new	173
\Gls: made robust	3.02 – 2012-05-21	
\glsadd: made robust	\@do@wrglossary: changed	
\glsaddall: made robust	\@glslocref to	
\GLSdesc: made robust	\theglsentrycounter	206
\Glsdesc: made robust	\@do@wrglossary: changed	
\glsdesc: made robust	\@do@wrglossary to test for	
\GLSdescplural: made robust	indexonlyfirst option; put old	
\Glsdescplural: made robust	\@do@wrglossary code into	
\glsdescplural: made robust	\@do@wrglossary	200
\glsfirst: made robust	\@gls@missingnumberlist: new	79
\GLSfirstplural: made robust	\@glswritefiles: added check	
\Glsfirstplural: made robust	for existence of token in case	
\glsfirstplural: made robust	\makeglossaries has been	
\glslink: made robust	omitted	198
\GLSname: made robust	\@printglossary: add a way to	
\Glsname: made robust	fetch current entry label . .	213

<code>\gls@defglossaryentry</code> : added numberlist element	97	<code>\gls@postinline</code> : replaced “.” with <code>\gls@postdescription</code>	272
<code>\gls@save@numberlist</code> : new	209	<code>index</code> : added check for <code>glsnogroupskip</code>	315
<code>\gls@wrglossary</code> : added check for glossary file defined	200	<code>list</code> : added check for <code>glsnogroupskip</code>	274
<code>\glsdisplaynumberlist</code> : new	172	<code>long</code> : added check for <code>glsnogroupskip</code>	278
<code>\glsentrycounter</code> : set default value	122	<code>long3col</code> : added check for <code>glsnogroupskip</code>	279
<code>\Glsentryfull</code> : fixed bug (replaced <code>\glsentryshortpl</code> with <code>\glsentryshort</code>)	172	<code>long4col</code> : added check for <code>glsnogroupskip</code>	281
<code>\glsentryfullpl</code> : fixed bug (replaced <code>\glsentryshort</code> with <code>\glsentryshortpl</code>)	172	<code>longragged</code> : added check for <code>glsnogroupskip</code>	289
<code>\glsentrynumberlist</code> : new	172	<code>longragged3col</code> : added check for <code>glsnogroupskip</code>	291
<code>\glsmoveentry</code> : new	99	<code>nogroupskip</code> : new	12
<code>\glsresetentrycounter</code> : new	227	<code>nopostdot</code> : new	11
<code>\ifglshaschildren</code> : new	65	<code>super</code> : added check for <code>glsnogroupskip</code>	300
<code>\ifglshasparent</code> : new	65	<code>super3col</code> : added check for <code>glsnogroupskip</code>	302
<code>indexonlyfirst</code> : new	29	<code>super4col</code> : added check for <code>glsnogroupskip</code>	304
<code>\makeglossaries</code> : added list parser	192	<code>superragged</code> : added check for <code>glsnogroupskip</code>	308
<code>\renewglossarystyle</code> : new	234	<code>superragged3col</code> : added check for <code>glsnogroupskip</code>	310
<code>savenumberlist</code> : new	10	<code>tree</code> : added check for <code>glsnogroupskip</code>	317
<code>\showglossaryentries</code> : fixed misspelt command	267	<code>treenoname</code> : added check for <code>glsnogroupskip</code>	318
<code>\SmallNewAcronymDef</code> : fixed broken short and long plural	260		
<code>ucmark</code> : new	12		
3.03 – 2012/09/21		3.04 – 2012-11-11	
General: allow title to set <code>toctitle</code>		<code>altlist</code> : replaced <code>\newline</code> with paragraph break	
<code>\@gls@sanitizesort</code> : new	23	3.04 – 2012-11-18	
<code>\@gls@setupsort@standard</code> : used <code>\@gls@sanitizesort</code>	14	General: Added check for doc package	
<code>\@printglossary</code> : allow title to override default <code>toctitle</code>	211	added <code>datatool</code> -base as a required package	
<code>altlongragged4col</code> : added check for <code>glsnogroupskip</code>	292	added local key	
<code>altsuperragged4col</code> : added check for <code>glsnogroupskip</code>	311	<code>\@do@wrglossary</code> : changed <code>\theglsentrycounter</code> back to <code>\@glslocref</code>	206
<code>alttree</code> : added check for <code>glsnogroupskip</code>	321	<code>\@do@esc@wrglossary</code> : modified to compensate for possible incorrect page number	204
<code>\glsinlinedescformat</code> : new	272	<code>\@gls@esc@sdq</code> : unsanitize <code>\gls@numberpage</code> , <code>\gls@alphpage</code> ,	
<code>\glsinlineemptydescformat</code> : new	272		
<code>\glsinlinenameformat</code> : new	272		
<code>\glsinlinepostchild</code> : new	272		
<code>\glsinlinesubdescformat</code> : new	273		
<code>\glsinlinesubnameformat</code> : new	273		

\gls@Alphpage and \gls@romanpage	125	\findrootlanguage: Obsoleted	60
\@print@glossary: Moved aux write to end of document to prevent unwanted whatsit occurring here.	213	3.07 – 2013-07-05 \@gls@link: fixed bug that failed to find entry in list	121
\gls@Alphpage: new	201	\glossarypreamble: modified to work with \setglossarypreamble	48
\gls@alphpage: new	201	\gls@docclearpage: added check for openright	51
\gls@disablepagerefexpansion: new	200	\glspostdescription: Added spacefactor code	11
\gls@numberpage: new	201	\GlsSetXdyCodePage: Added check for fontspec	60
\gls@protected@pagefmts: new	200	\SetDescriptionAcronymDisplayStyle: now using \glsdoparenifnotempty	259
\gls@romanpage: new	201	\setglossarypreamble: new	48
\glsdefmain: added check for doc package	17	3.08a – 2013-08-30 altlist: updated altlist style to use \glossentry and \subglossentry	275
\glsorg@endtheglossary: new	4	inline: updated inline style to use \glossentry and \subglossentry	271
\glsorg@theglossary: new	4	list: updated list style to use \glossentry and \subglossentry	274
\PrintChanges: new	4	listdotted: updated listdotted style to use \glossentry and \subglossentry	276
3.05 – 2013-04-21 \@do@esc@wrglossary: add Roman case. Fixed bugs in the else statements	204	3.08a – 2013-09-28 General: added nogroupskip key to \printglossary	225
\@gls@link: added check for “nohypertypes”	121	removed definition of \@glossaryentryfield	357
\gls@protected@pagefmts: added Roman to list	200	removed definition of \@glossarysubentryfield	357
\gls@Romanpage: new	201	\@glo@storeentry: no longer need to check for special characters in any of the fields other than sort	100
\glsgetgrouplabel: fixed bug (typo in \equal)	232	updated for \glossentry	100
\glsmcols: new	294	\@glossaryentryfield: switched to \glossentry	99
mcolalttree: replaced ‘2’ with \glsmcols	298	\@glossarysubentryfield: switched to \subglossentry	99
mcolindex: replaced ‘2’ with \glsmcols	294	\@gls@sanitizedesc: deprecated	22
mcolindexspannav: replaced ‘2’ with \glsmcols	295	altlongragged4col: updated to use \glossentry and \subglossentry	292
mcoltree: replaced ‘2’ with \glsmcols	295		
mcoltreenoname: replaced ‘2’ with \glsmcols	297		
mcoltreespannav: replaced ‘2’ with \glsmcols	296		
\nopostdesc: made robust	45		
3.05 – 2013/04/21 \@gls@nohyperlist: new	20		
\GlsDeclareNoHyperList: new	20		
nohypertypes: new	21		
3.06 – 2013/06/17 \@xdy@main@language: Changed back to using \language	31		

alttree: updated to use		superragged: updated to use	
\glossentry and		\glossentry and	
\subglossentry	320	\subglossentry	307
\compatibleglossentry: new ..	229	tree: updated to use	
\compatiblesubglossentry:		\glossentry and	
new	230	\subglossentry	316
\glossaryentryfield:		3.09a – 2013-10-09	
deprecated	230	\@gls@assign@symbolplural@field:	
\glossarystyle: deprecated ..	233	new	23
\glossentry: new	229	\@gls@default@value: new ..	74
\Glossentrydesc: new	230	\Glsentrydesc: made robust ..	166
\glossentrydesc: new	230	\Glsentrydescplural: made	
\Glossentryname: new	229	robust	166
\glossentryname: new	229	\Glsentryfirst: made robust ..	167
\Glossentrysymbol: new	230	\Glsentryfirstplural: made	
\glossentrysymbol: new	230	robust	168
\gls@assign@desc@field: new ..	22	\Glsentryfull: made robust ..	172
\gls@assign@descplural@field:		\Glsentryfullpl: made robust ..	172
new	23	\Glsentrylong: made robust ..	171
\gls@assign@field: new	81	\Glsentrylongpl: made robust ..	171
\gls@ifnotmeasuring: new ..	102	\Glsentryname: made robust ..	164
\glsaddallunused: new	175	\Glsentryplural: made robust ..	167
\glsexpandfields: new	81	\Glsentryshort: made robust ..	171
\glsnoexpandfields: new	81	\Glsentryshortpl: made robust ..	171
\glssee: made robust	208	\Glsentrysymbol: made robust ..	167
\glsseeformat: made robust ..	208	\Glsentrysymbolplural: made	
\glsseeitem: made robust	209	robust	167
\glsseelist: made robust	208	\Glsentrytext: made robust ..	166
\ifglsdessuppressed: new ..	66	\Glsentryuseri: made robust ..	169
\ifglshasdesc: new	65	\Glsentryuserii: made robust ..	169
\ifglshassymbol: new	66	\Glsentryuseriii: made robust ..	169
index: added paragraph break at		\Glsentryuseriv: made robust ..	170
end of environment	314	\Glsentryuserv: made robust ..	170
updated to use \glossentry		\Glsentryuservi: made robust ..	170
and \subglossentry	315	\glstextup: new	239
long: updated to use		\ifglshassymbol: changed test	
\glossentry and		to check for	
\subglossentry	278	\@gls@default@symbol	66
longragged: updated to use		3.10a – 2013-09-28	
\glossentry and		\gls@assign@type@field: new ..	22
\subglossentry	289	3.10a – 2013-10-13	
longragged3col: updated to use		\@gls@keymap: new	84
\glossentry and		\@gls@provide@newglossary:	
\subglossentry	290	new	69
\setglossarystyle: new	233	\@gls@writedef: new	84
\setglossentrycompatibility:		\@glsdefaultplural: Obsolete ..	78
new	230	\@glsnodesc: new	78
\subglossentry: new	231	\@print@glossary: Added	
		providecommand code to aux	
		file	214

<code>\gls@defglossaryentry:</code>		
Changed to using		
<code>\@gls@default@value</code>	93	
new	93	
<code>\glswritedefhook:</code> new	92	
<code>\makeglossaries:</code> Added		
providecommand code to aux		
file	191	
<code>\new@glossaryentry:</code> new	82	
<code>\ns@newglossary:</code> added		
<code>\@gls@provide@newglossary</code>	71	
3.11a – 2013-10-15		
General: added <code>\glslabel</code> ,		
<code>\glsifplural</code> , <code>\glscapscase</code> ,		
<code>\glsinsert</code> and		
<code>\glscustomtext</code>	155–162	
changed to just use		
<code>\Glsentrydescplural</code>	146	
changed to just use		
<code>\glsentrydescplural</code>	146	
changed to just use		
<code>\Glsentrydesc</code>	145	
changed to just use		
<code>\glsentrydesc</code>	145	
changed to just use		
<code>\Glsentryfirstplural</code>	143	
changed to just use		
<code>\glsentryfirstplural</code>	143	
changed to just use		
<code>\Glsentryfirst</code>	141	
changed to just use		
<code>\glsentryfirst</code>	141	
changed to just use		
<code>\Glsentryname</code>	144	
changed to just use		
<code>\glsentryname</code>	144	
changed to just use		
<code>\Glsentryplural</code>	142	
changed to just use		
<code>\glsentryplural</code>	142	
changed to just use		
<code>\Glsentrysymbolplural</code>	148	
changed to just use		
<code>\glsentrysymbolplural</code>	148	
changed to just use		
<code>\Glsentrysymbol</code>	147	
changed to just use		
<code>\glsentrysymbol</code>	147	
Changed to just use		
<code>\Glsentrytext</code>	140	
changed to just use		
<code>\glsentrytext</code>	140	
changed to just use		
<code>\Glsentryuseriii</code>	151	
changed to just use		
<code>\glsentryuseriii</code>	151	
changed to just use		
<code>\Glsentryuserii</code>	150	
changed to just use		
<code>\glsentryuserii</code>	150	
changed to just use		
<code>\Glsentryuseriv</code>	152	
changed to just use		
<code>\glsentryuseriv</code>	152	
changed to just use		
<code>\Glsentryuseri</code>	149	
changed to just use		
<code>\glsentryuseri</code>	149	
changed to just use		
<code>\Glsentryuservi</code>	154	
changed to just use		
<code>\glsentryuservi</code>	154	
changed to just use		
<code>\Glsentryuserv</code>	153	
changed to just use		
<code>\glsentryuserv</code>	153	
Now requires <code>textcase</code>	3	
<code>\@ACRlong:</code> added <code>\glslabel</code> ,		
<code>\glsifplural</code> , <code>\glscapscase</code> ,		
<code>\glsinsert</code> and		
<code>\glscustomtext</code>	352	
<code>\@ACRshort:</code> added <code>\glslabel</code> ,		
<code>\glsifplural</code> , <code>\glscapscase</code> ,		
<code>\glsinsert</code> and		
<code>\glscustomtext</code>	351	
<code>\@Acrlong:</code> added <code>\glslabel</code> ,		
<code>\glsifplural</code> , <code>\glscapscase</code> ,		
<code>\glsinsert</code> and		
<code>\glscustomtext</code>	352	
<code>\@Acrshort:</code> added <code>\glslabel</code> ,		
<code>\glsifplural</code> , <code>\glscapscase</code> ,		
<code>\glsinsert</code> and		
<code>\glscustomtext</code>	350	
<code>\@GLS@:</code> add <code>\glslabel</code> ,		
<code>\glsifplural</code> , <code>\glscapscase</code> ,		
<code>\glscustomtext</code> and		
<code>\glsinsert</code>	136	
change to using <code>\glsentryfmt</code>		
style commands	136	

removed \MakeUppercase (now moved to \glentryfmt) ..	136	\@glsp10: add \glslabel,	
\@GLSpl: add \glslabel,		\glslifplural, \glscapscase,	
\glslifplural, \glscapscase,		\glscustomtext and	
\glinsert	138	\glinsert	136
change to using \glentryfmt		change to using \glentryfmt	
style commands	138	style commands	136
removed \MakeUppercase as		acronymlists: replaced	
now dealt with in		\@addtoacronymlists with	
\glentryfmt	138	\DeclareAcronymList	20
\@Gls@: add \glslifplural,		compatible-3.07: new	36
\glscapscase,		\defglssdisplay: obsoleted	117
\glscustomtext and		\defglssdisplayfirst: obsoleted	117
\glinsert	135	\defglssentryfmt: new	69
change to using \glentryfmt		\forglssentries: replaced \ifx	
style commands	135	with \ifdefempty	62
removed \makefirsttuc (now		\glss@assign@desc: new	92
dealt with in \glentryfmt)	135	\glss@defglssaryentry: Fixed	
\@GLSpl0: add \glslifplural,		default counter if none	
\glscapscase,		supplied	96
\glscustomtext and		\glss@doentryfmt: new	69
\glinsert	137	\glssdisplay: obsoleted	117
change to using \glentryfmt		\glssdisplayfirst: obsoleted ..	117
style commands	137	\glssgenentryfmt: new	112
removed \makefirsttuc (now		\glssgetgrouptitle: Added check	
dealt with in \glentryfmt)	137	in case non-Latin alphabet in	
\@acrlong: added \glslabel,		use	232
\glslifplural, \glscapscase,		\glssglossarymark: replaced	
\glinsert and		\MakeUppercase with	
\glscustomtext	351	\mfirsttucMakeUppercase ..	49
\@acrshort: added \glslabel,		\glssnavigation: switched to	
\glslifplural, \glscapscase,		using \@glss@getgrouptitle	270
\glinsert and		\ifglsshasdesc: replaced	
\glscustomtext	350	\ifdefempty with	
\@glss@: add \glslabel,		\ifcseempty	65
\glslifplural, \glscapscase,		\ifglsshaslong: new	66
\glscustomtext and		\ifglsshasshort: new	66
\glinsert	134	\ifglsshasymbol: replaced	
change to using \glentryfmt		\ifdefempty with	
style commands	134	\ifcseempty	66
\@glss@noexpand@fields: Fixed		\ifglssused: replaced	
bug expand replaced with		\ifthenelse with \ifbool ..	63
noexpand	79	\longnewglossaryentry: new ..	92
\@glssdisp: add \glslabel,		\ns@newglossary: replaced	
\glslifplural, \glscapscase,		\glssdisplay and	
\glscustomtext and		\glssdisplayfirst with	
\glinsert	139	\glssentryfmt	71
change to using \glentryfmt		numbers: new	38
style commands	139	\SetCustomDisplayStyle:	
		updated to use	
		\defglssentryfmt	261

\SetDefaultAcronymDisplayStyle:	\gls@assign@desc@field:
changed to use	changed to use
\defglsentryfmt 258	\glsetnoexpandfield 22
\SetDescriptionAcronymDisplayStyle:	\gls@assign@descplural@field:
updated to use	changed to use
\defglsentryfmt 259	\glsetnoexpandfield 23
\SetDescriptionDUAAcronymDisplayStyle:	\gls@assign@name@field:
updated to use	changed to use
\defglsentryfmt 259	\glsetnoexpandfield 23
\SetDescriptionFootnoteAcronymDisplayStyle:	\gls@assign@type@field:
updated to use	changed to use
\defglsentryfmt 258	\glsetexpandfield 22
\SetDUADisplayStyle: updated	\gls@checkseeallowed: new .. 75
to use \defglsentryfmt .. 260	\glsaddallunused: set default to
\SetFootnoteAcronymDisplayStyle:	\@glo@types 175
updated to use	\Glsentryfull: changed to use
\defglsentryfmt 259	\acrfullformat 172
\SetSmallAcronymDisplayStyle:	\glsetentryfull: changed to use
updated to use	\acrfullformat 172
\defglsentryfmt 260	\Glsentryfullpl: changed to use
\setupglossaries: new 40	\acrfullformat 172
\showglolong: new 265	\glsetentryfullpl: changed to use
\showgloshort: new 265	\acrfullformat 172
symbols: new 38	\glsglossarymark: renamed
3.12a – 2013-10-16	\glossarymark to
\gls@defglossaryentry: added	\glsglossarymark to avoid
\glslabel 93	conflict with memoir 49
\glsaddkey: new 86	\glsprestandardsort: new ... 13
3.13a – 2013-11-05	\glsetexpandfield: new 22
\@gls@assign@symbol@field:	\glsetnoexpandfield: new .. 22
changed to use	long: switched to
\glsetnoexpandfield 23	\tabularnewline 278
\@gls@assign@symbolplural@field:	long3col: switched to
changed to use	\tabularnewline 279
\glsetnoexpandfield 23	long3colheader: switched to
\@gls@link: removed \relax .. 122	\tabularnewline 280
\@gls@notranslatorhook: new . 27	long3colheaderborder: switched
\@gls@setupsort@standard:	to \tabularnewline 280
moved \@gls@santizesort to	long4col: switched to
\glsprestandardsort 14	\tabularnewline 281
altsuper4colheader: switched to	long4colheader: switched to
\tabularnewline 305	\tabularnewline 281
altsuper4colheaderborder:	longheader: switched to
switched to	\tabularnewline 278
\tabularnewline 306	longheaderborder: switched to
\glossarysection: changed	\tabularnewline 279
\glossarymark to	see: added
\glsglossarymark 49	\gls@checkseeallowed 75
	super: switched to
	\tabularnewline 300

super3col: switched to		\@acrfull: now using	
\tabularnewline	302	\acrfullfmt	241
super3colheader: switched to		\@gls@indexdef: new	39
\tabularnewline	303	\@gls@numbersdef: new	38
super4col: switched to		\@gls@symbolsdef: new	38
\tabularnewline	304	\ACRfullfmt: new	240
super4colheader: switched to		\Acrfullfmt: new	240
\tabularnewline	304	\acrfullfmt: new	240
super4colheaderborder:		\ACRfullplfmt: new	241
switched to		\Acrfullplfmt: new	241
\tabularnewline	305	\acrfullplfmt: new	240
superheader: switched to		\acronymentry: new	244
\tabularnewline	301	dua: new	251
superheaderborder: switched to		dua-desc: new	253
\tabularnewline	301	footnote: new	253
ucmark: added check for memoir	12	footnote-desc: new	255
3.14a – 2013-11-12		footnote-sc: new	254
General: new	324	footnote-sc-desc: new	255
\@glswritefiles: renamed		footnote-sm: new	255
\glswritefiles to		footnote-sm-desc: new	256
\@glswritefiles and used		\Genacrfullformat: new	116
“savewrites” option to set		\genacrfullformat: new	116
\glswritefiles	198	\GenericAcronymFields: new	244
acronyms: new	18	\Genplacrfullformat: new	117
compatible-2.07: added check		\genplacrfullformat: new	116
for 2.07 before setting 3.07		\Glsentryfull: bug fix: added	
compatibility	36	missing \acronymfont	172
\gls@defglossaryentry: added		\Glsentryfull: bug fix: added	
check for existence of default		missing \acronymfont	172
glossary	94	\Glsentryfullpl: bug fix: added	
set the default for firstplural to		missing \acronymfont	172
be the value of plural	96	\Glsentryfullpl: bug fix: added	
\longprovideglossaryentry:		missing \acronymfont	172
new	93	\glsacrfmt: new	114
notranslate: new	28	\GlsUseAcrEntryDisplayStyle:	
\provideglossaryentry: new	81	new	246
xindygloss: new	32	\GlsUseAcrStyleDefs: new	246
4.0 – 2013-11-14		index: new	38
\gls@defglossaryentry: added		long-sc-short: new	248
check for first key	96	long-sc-short-desc: new	249
super: fixed typo in		long-short: new	246
\subglossentry		long-short-desc: new	249
(\glossentrydesc)	300	long-sm-short: new	248
4.01 – 2013-11-16		long-sm-short-desc: new	250
General: fixed non-value options		\newacronymstyle: new	245
so that they can be passed to		numberedsection: added nameref	
document class	9	option	8
4.02 – 2013-12-05		sanitize: fixed bug that caused	
General: Removed		an error here	27
\acronymfont	159–162	sc-short-long: new	248

sc-short-long-desc: new	250	moved \@gls@link@opts and \@gls@link@label to \@gls@link	121
\setacronymstyle: new	245	\@gls@writedef: added \glsdetoklabel	84
\SetGenericNewAcronym: new . .	243	\@glsdisp: removed \glslabel (defined in \@gls@link) . .	139
short-long: new	247	\@glspl@: removed \glslabel (defined in \@gls@link) . .	136
short-long-desc: new	250	\@printglossary: added \glsdetoklabel	213
sm-short-long: new	249	\compatibleglossentry: added \glsdetoklabel	332
sm-short-long-desc: new	251	\compatiblesubglossentry: added \glsdetoklabel . . .	332
xindynoglsnumbers: new	32	dua: fixed bug in \acrfullfmt .	252
4.02 – 2013-13-05		fixed bug in \Acrfullplfmt .	252
\makeglossaries: made preamble only	193	fixed bug in \acrfullplfmt .	252
4.03 – 2014-01-17		redefined to use accessibility information	362
General: changed default to \@empty instead of \@relax . .	36	dua-desc: commented spurious EOL	253
4.03 – 2014-01-20		redefined to use accessibility information	364
General: removed \glslabel (defined in \@gls@link) . .	155	footnote: redefined to use accessibility information . .	365
\@do@esc@wrglossary: added \glsdetoklabel	205	footnote-desc: redefined to use accessibility information . .	367
\@do@noesc@wrglossary: added \glsdetoklabel	203	footnote-sc: redefined to use accessibility information . .	366
\@ACRlong: removed \glslabel (defined in \@gls@link) . .	352	footnote-sc-desc: redefined to use accessibility information	367
\@ACRshort: removed \glslabel (defined in \@gls@link) . .	351	footnote-sm: redefined to use accessibility information . .	367
\@Acrlong: removed \glslabel (defined in \@gls@link) . .	352	footnote-sm-desc: redefined to use accessibility information	367
\@Acrshort: removed \glslabel (defined in \@gls@link) . .	350	\Genacrfullformat: redefined to use accessibility information	350
\@GLS@: removed \glslabel (defined in \@gls@link) . .	136	\genacrfullformat: redefined to use accessibility information	349
\@GLSpl: removed \glslabel (defined in \@gls@link) . .	138	\Genplacrfullformat: redefined to use accessibility information	350
\@Gls@: removed \glslabel (defined in \@gls@link) . .	135	\genplacrfullformat: redefined to use accessibility information	350
\@Gls@entry@field: new	163	\glossentryname: added \glsdetoklabel	229
\@Glspl@: removed \glslabel (defined in \@gls@link) . .	137	\gls@defglossaryentry: added \glsdetoklabel	93
\@acrlong: removed \glslabel (defined in \@gls@link) . .	351		
\@acrshort: removed \glslabel (defined in \@gls@link) . .	350		
\@gls@: removed \glslabel (defined in \@gls@link) . .	134		
\@gls@access@display: new . .	341		
\@gls@entry@field: new	162		
\@gls@fetchfield: new	85		
\@gls@field@link: new	139		
\@gls@link: added \glsdetoklabel	121		

replaced #1 with \@glo@label	94	\glslocalunset: added	
replaced \ifthenelse with		\glsdetoklabel	104
\ifdefequal	95	\glsmoveentry: added	
\glsadd: added \glsdetoklabel	174	\glsdetoklabel	99
\glsaddkey: switched to using		replaced \ifthenelse with	
\@gls@field@link	87	\ifdefequal	99
\glsdetoklabel: new	63	\glsrefentry: added	
\glsdisplaynumberlist: added		\glsdetoklabel	227
\glsdetoklabel	173	\glsreset: added	
\glsdoifexistsorwarn: new	64	\glsdetoklabel	103
\glsentryaccess: switched to		\glsseelist: added	
using \@gls@entry@field	337	\expandafter commands	209
\glsentrydescaccess: switched		\glsstepentry: added	
to using \@gls@entry@field	338	\glsdetoklabel	227
\glsentrydescpluralaccess:		\glsstepsubentry: added	
switched to using		\glsdetoklabel	227
\@gls@entry@field	338	\glsunset: added	
\glsentryfirstaccess: switched		\glsdetoklabel	103
to using \@gls@entry@field	338	\ifglsdescsuppressed: added	
\glsentryfirstplural: added		\glsdetoklabel	66
\glsdetoklabel	168	fixed typo	66
\glsentrylongaccess: switched		\ifglsentryexists: added	
to using \@gls@entry@field	339	\glsdetoklabel	63
\glsentrylongpluralaccess:		\ifglschaschildren: added	
switched to using		\glsdetoklabel	65
\@gls@entry@field	339	\ifglschasdesc: added	
\glsentrypluralaccess:		\glsdetoklabel	65
switched to using		\ifglschasfield: new	67
\@gls@entry@field	338	\ifglschaslong: added	
\glsentryshortaccess: switched		\glsdetoklabel	66
to using \@gls@entry@field	338	\ifglschasparent: added	
\glsentryshortpluralaccess:		\glsdetoklabel	65
switched to using		\ifglschasshort: added	
\@gls@entry@field	338	\glsdetoklabel	66
\glsentrysymbolaccess:		\ifglschassymbol: added	
switched to using		\glsdetoklabel	66
\@gls@entry@field	338	replaced \ifcsempy with	
\glsentrysymbolpluralaccess:		\ifdefempty and replaced	
switched to using		\ifx with \ifdefequal	66
\@gls@entry@field	338	\ifglsused: added	
\glsentrytextaccess: switched		\glsdetoklabel	63
to using \@gls@entry@field	337	long-sc-short-desc: redefined	
\glsacacfmt: redefined to use		to use accessibility	
accessibility information	347	information	360
\glsacentryfmt: redefined to		long-short: redefined to use	
use accessibility information	345	accessibility information	359
\gshyperlink: added		long-short-desc: redefined to	
\glsdetoklabel	173	use accessibility information	360
\glslocalreset: added			
\glsdetoklabel	103		

long-sm-short-desc: redefined to use accessibility information	361	\showgloplural: added \glsdetoklabel	262
\renewacronymstyle: new	246	\showglopluralaccess: added \glsdetoklabel	369
sc-short-long-desc: redefined to use accessibility information	361	\showgloshort: added \glsdetoklabel	265
short-long: commented spurious EOL	248	\showgloshortaccess: added \glsdetoklabel	370
redefined to use accessibility information	359	\showgloshortpluralaccess: added \glsdetoklabel . . .	370
short-long-desc: redefined to use accessibility information	361	\showglosort: added \glsdetoklabel	264
\showglocounter: added \glsdetoklabel	263	\showglosymbol: added \glsdetoklabel	264
\showglodesc: added \glsdetoklabel	264	\showglosymbolaccess: added \glsdetoklabel	369
\showglodescaccess: added \glsdetoklabel	369	\showglosymbolplural: added \glsdetoklabel	265
\showglodescplural: added \glsdetoklabel	264	\showglosymbolpluralaccess: added \glsdetoklabel . . .	369
\showglodescpluralaccess: added \glsdetoklabel . . .	369	\showglotext: added \glsdetoklabel	262
\showglofirst: added \glsdetoklabel	262	\showglotextaccess: added \glsdetoklabel	369
\showglofirstaccess: added \glsdetoklabel	369	\showglotype: added \glsdetoklabel	262
\showglofirsttpl: added \glsdetoklabel	262	\showglouseri: added \glsdetoklabel	263
\showglofirstpluralaccess: added \glsdetoklabel . . .	369	\showglouserii: added \glsdetoklabel	263
\showgloflag: added \glsdetoklabel	265	\showglouseriii: added \glsdetoklabel	263
\showgloindex: added \glsdetoklabel	265	\showglouseriv: added \glsdetoklabel	263
\showglolevel: added \glsdetoklabel	262	\showglouserv: added \glsdetoklabel	263
\showglolong: added \glsdetoklabel	265	\showglouservi: added \glsdetoklabel	264
\showglolongaccess: added \glsdetoklabel	370	sm-short-long-desc: redefined to use accessibility information	362
\showglolongpluralaccess: added \glsdetoklabel . . .	370	4.04 – 2014-03-04 \@gls@getcounterprefix: added warning if no prefix can be formed	207
\showgloname: added \glsdetoklabel	264	4.04 – 2014-03-06 General: added sort key to printgloss group	226
\showglonameaccess: added \glsdetoklabel	369	\@@gls@noidx@nosanitizesort: new	24
\showgloparent: added \glsdetoklabel	261		

\@gls@noidx@sanitizesort:		\gls@defglossaryentry:	
new	23	replaced \ifx with	
\@gls@nosanitizesort: new ..	23	\ifdefvoid	98
\@gls@sanitizesort: new ...	23	\glsnoidxdisplayloc: new ...	223
\@glo@addchildren: new	215	\glsnoidxdisplaylocclishandler:	
\@glo@do@sortentries: new ..	215	new	223
\@glo@grabfirst: new	221	\glsnoidxloclist: new	222
\@glo@sortedinsert: new	216	\glsnoidxlocclishandler: new	223
\@glo@sortentries: new	214	\glsnoidxstripaccents: new ..	24
\@glo@sorthandler@case: new ..	217	\makeglossaries: Moved	
\@glo@sorthandler@letter:		definition of \glswrite to	
new	217	\makeglossaries	191
\@glo@sorthandler@nocase:		\makenoidxglossaries: new ..	193
new	217	\printglossary: changed to use	
\@glo@sorthandler@word: new ..	216	new \@printglossary	210
\@glo@sortmacro@case: new ..	218	\printnoidxglossaries: new ..	211
\@glo@sortmacro@def: new ...	219	\printnoidxglossary: new ...	211
\@glo@sortmacro@def@do: new ..	219	\showgloloclist: new	265
\@glo@sortmacro@letter: new ..	218	\warn@noprintglossary:	
\@glo@sortmacro@nocase: new ..	218	Activate warning in	
\@glo@sortmacro@standard:		\makeglossaries	210
new	218	\writeist: checked for definition	
\@glo@sortmacro@use: new ...	219	of \glswrite	178, 182
\@glo@sortmacro@word: new ..	217	4.06 – 2014-03-12	
\@gls@noidx@do: new	221	General: added	
\@gls@noidx@getgrouptitle:		\glsifhyper	155–162
new	232	\@GLS@: added \glsifhyper ...	136
\@gls@noref@warn: new	197	\@GLSpl: added \glsifhyper ..	138
\@gls@reference: new	223	\@Gls@: added \glsifhyper ...	135
\@gls@warnonglossdefined: new	22	\@Glspl@: added \glsifhyper ..	137
\@gls@warnonthe glossdefined:		\@gls@: added \glsifhyper ...	134
new	22	\@gls@numbersdef: added hook	
\@no@makeglossaries: new ...	197	to set toc title	38
\@print@glossary: new	213	\@gls@symbolsdef: added hook	
\@print@noidx@glossary: new ..	219	to set toc title	38
\@print@gloss@setsort: new ..	211	\@glsdisp: added \glsifhyper	139
\@printglossary: new	211	\@glspl@: added \glsifhyper ..	136
alttree: moved hangindent and		acronym: added hook to set toc	
parindent assignments outside		title	18
level test	320	acronyms: added hook to set toc	
\compatibleglossentry: changed		title	18
\newcommand to \def as is		\glsdefmain: added hook to set	
may or may not be defined	332	toc title	17
\compatiblesubglossentry:		4.07 – 2014-04-04	
changed \newcommand to \def		\@glossarysection: added	
as is may or may not be		optional argument when using	
defined	332	unstarred version	50
\glo@grabfirst: new	220	\@gls@noidx@do: added \global	
		in case it's used in a	
		tabular-like style	221

<code>\Acrfullplfmt</code> : fixed no case		removed <code>\@sGLSsymbol</code>	147
change bug	241	removed <code>\@sGLssymbol</code>	147
<code>\glsletentryfield</code> : new	163	removed <code>\@sGLStext</code>	140
4.08 – 2014-07-30		removed <code>\@sGlstext</code>	140
General: added entrycounter		removed <code>\@sGlstext</code>	140
option to <code>printgloss</code> family	225	removed <code>\@sGLSuseriii</code>	151
added nopostdot option to		removed <code>\@sGlsuseriii</code>	151
<code>printgloss</code> family	225	removed <code>\@sGlsuseriii</code>	151
added subentrycounter option		removed <code>\@sGLSuserii</code>	150
to <code>printgloss</code> family	225	removed <code>\@sGlsuserii</code>	150
explicitly initialise hyper key	118	removed <code>\@sGlsuserii</code>	150
moved <code>\glsifhyper</code>	155–162	removed <code>\@sGLSuseriv</code>	152
removed <code>\@sACRlongpl</code>	162	removed <code>\@sGlsuseriv</code>	152
removed <code>\@sAcrlongpl</code>	161	removed <code>\@sGlsuseriv</code>	152
removed <code>\@sacrlongpl</code>	160	removed <code>\@sGLSuseri</code>	149
removed <code>\@sACRlong</code>	160	removed <code>\@sGlsuseri</code>	149
removed <code>\@sAcrlong</code>	159	removed <code>\@sGlsuseri</code>	149
removed <code>\@sacrlong</code>	158	removed <code>\@sGLSuservi</code>	154
removed <code>\@sACRshortpl</code>	158	removed <code>\@sGlsuservi</code>	154
removed <code>\@sAcrshortpl</code>	157	removed <code>\@sGlsuservi</code>	154
removed <code>\@sacrshortpl</code>	156	removed <code>\@sGLSuserv</code>	153
removed <code>\@sACRshort</code>	156	removed <code>\@sGlsuserv</code>	153
removed <code>\@sAcrshort</code>	155	removed <code>\@sGlsuserv</code>	153
removed <code>\@sacrshort</code>	155	removed <code>\@sGLS</code>	135
removed <code>\@sgls@link</code>	120	removed <code>\@sGls</code>	135
removed <code>\@sGLSdescplural</code>	146	removed <code>\@sgls</code>	134
removed <code>\@sGlsdescplural</code>	146	removed <code>\@thirdofthree</code>	
removed <code>\@sglsdescplural</code>	146	(defined in kernel)	133
removed <code>\@sGLSdesc</code>	145	removed <code>sPGLS</code>	330
removed <code>\@sGlsdesc</code>	145	removed <code>sPgls</code>	328
removed <code>\@sglsdesc</code>	145	removed <code>spgls</code>	327
removed <code>\@sglsdisp</code>	138	removed <code>sPGLSpl</code>	330
removed <code>\@sGLSfirstplural</code>	143	removed <code>sPglspl</code>	329
removed <code>\@sGlsfirstplural</code>	143	removed <code>spglspl</code>	328
removed <code>\@sglsfirstplural</code>	143	<code>\@ACRlong</code> : added	
removed <code>\@sGLSfirst</code>	141	<code>\do@gls@link@checkfirsthyper</code>	
removed <code>\@sGlsfirst</code>	141	352
removed <code>\@sglsfirst</code>	141	<code>\@ACRshort</code> : added	
removed <code>\@sGLSname</code>	144	<code>\do@gls@link@checkfirsthyper</code>	
removed <code>\@sGlsname</code>	144	351
removed <code>\@sglsname</code>	144	<code>\@Acrlong</code> : added	
removed <code>\@sGLSplural</code>	142	<code>\do@gls@link@checkfirsthyper</code>	
removed <code>\@sGlsplural</code>	142	351
removed <code>\@sglsplural</code>	142	<code>\@Acrshort</code> : added	
removed <code>\@sGLSpl</code>	138	<code>\do@gls@link@checkfirsthyper</code>	
removed <code>\@sGlspl</code>	137	350
removed <code>\@sglspl</code>	136	<code>\@GLS@</code> : moved <code>\glsifhyper</code>	136
removed <code>\@sGLSsymbolplural</code>	148	moved check for first use to	
removed <code>\@sGLssymbolplural</code>	148	<code>\@gls@link</code>	136
removed <code>\@sGLssymbolplural</code>	148		

\GLSpl: moved \glsifhyper ..	138	\ACRfullpl: removed	
moved check for first use to		\s@ACRfullpl	242
\@gls@link	138	switched to using	
\Gls@: moved \glsifhyper ...	135	\@gls@hyp@opt	242
moved check for first use to		\Acrfullpl: removed	
\@gls@link	135	\s@Acrfullpl	242
\Glspl@: moved \glsifhyper .	137	switched to using	
moved check for first use to		\@gls@hyp@opt	242
\@gls@link	137	\acrfullpl: removed	
\@acrlong: added		\s@acrfullpl	242
\do@gls@link@checkfirsthyper		switched to using	
.....	351	\@gls@hyp@opt	242
\@acrshort: added		\ACRlong: switched to using	
\do@gls@link@checkfirsthyper		\@gls@hyp@opt	160
.....	350	\Acrlong: switched to using	
\@closegls: new	188	\@gls@hyp@opt	159
\@gls@: moved \glsifhyper ...	134	\acrlong: switched to using	
moved check for first use to		\@gls@hyp@opt	158
\@gls@link	134	\ACRlongpl: switched to using	
\@gls@automake: new	188	\@gls@hyp@opt	162
\@gls@doautomake: new	35	\Acrlongpl: switched to using	
\@gls@field@link: added		\@gls@hyp@opt	161
assignment of		\acrlongpl: switched to using	
\do@gls@link@checkfirsthyper		\@gls@hyp@opt	160
.....	139	\ACRshort: switched to using	
\@gls@forbidtextext: new	69	\@gls@hyp@opt	156
\@gls@hyp@opt: new	119	\Acrshort: switched to using	
\@gls@link: removed		\@gls@hyp@opt	155
redundancy	121	\acrshort: switched to using	
renamed \gls@type to		\@gls@hyp@opt	154
\glstype	121	\ACRshortpl: switched to using	
\@gls@link@checkfirsthyper:		\@gls@hyp@opt	158
new	120	\Acrshortpl: switched to using	
\@glsdisp: moved \glsifhyper	139	\@gls@hyp@opt	157
moved check for first use to		\acrshortpl: switched to using	
\@gls@link	139	\@gls@hyp@opt	156
\@glspl@: moved \glsifhyper .	136	altlongragged4col: fixed bug	
moved check for first use to		that displayed description	
\@gls@link	136	instead of symbol	292
\@ignored@glossaries: new ..	73	automake: new	34
\ACRfull: removed \s@ACRfull	241	\forallacronyms: new	61
switched to using		\GLS: switched to using	
\@gls@hyp@opt	241	\@gls@hyp@opt	135
\Acrfull: removed \s@Acrfull	241	\Gls: switched to using	
switched to using		\@gls@hyp@opt	134
\@gls@hyp@opt	241	\gls: switched to using	
\acrfull: removed \s@acrfull	241	\@gls@hyp@opt	134
switched to using		\gls@defglossaryentry: added	
\@gls@hyp@opt	241	check for ignored glossary ..	94
		\gls@istfilebase: new	46

\glsaddkey: removed		
\@sGLS@user@<key>	88	
removed \@sGls@user@<key>	88	
removed \@sgls@user@<key>	87	
switched to using		
\@gls@hyp@opt	87, 88	
\GLSdesc: switched to using		
\@gls@hyp@opt	145	
\Glsdesc: switched to using		
\@gls@hyp@opt	145	
\glsdesc: switched to using		
\@gls@hyp@opt	144	
\GLSdescplural: switched to		
using \@gls@hyp@opt	146	
\Glsdescplural: switched to		
using \@gls@hyp@opt	146	
\glsdescplural: switched to		
using \@gls@hyp@opt	145	
\glsdisablehyper: added		
\KV@glslink@hyperfalse to		
definition	133	
\glsdisp: switched to using		
\@gls@hyp@opt	138	
\glsdohyperlink: new	132	
\glsdohypertarget: new	132	
\glsenablehyper: added		
\KV@glslink@hypertrue to		
definition	133	
\GLSfirst: switched to using		
\@gls@hyp@opt	141	
\Glsfirst: switched to using		
\@gls@hyp@opt	141	
\glsfirst: switched to using		
\@gls@hyp@opt	141	
\GLSfirstplural: switched to		
using \@gls@hyp@opt	143	
\Glsfirstplural: switched to		
using \@gls@hyp@opt	143	
\glsfirstplural: switched to		
using \@gls@hyp@opt	142	
\glsifhyper: deprecated	119	
\glslink: switched to using		
\@gls@hyp@opt	120	
\glslinkcheckfirsthyperhook:		
new	121	
\glslinkvar: new	119	
\GLSname: switched to using		
\@gls@hyp@opt	144	
\Glsname: switched to using		
\@gls@hyp@opt	144	
\glsname: switched to using		
\@gls@hyp@opt	143	
\GLSpl: switched to using		
\@gls@hyp@opt	138	
\Glspl: switched to using		
\@gls@hyp@opt	137	
\glspl: switched to using		
\@gls@hyp@opt	136	
\GLSplural: switched to using		
\@gls@hyp@opt	142	
\Glsplural: switched to using		
\@gls@hyp@opt	142	
\glsplural: switched to using		
\@gls@hyp@opt	141	
\glsspace: new	240	
\GLSsymbol: switched to using		
\@gls@hyp@opt	147	
\Glsymbol: switched to using		
\@gls@hyp@opt	147	
\glssymbol: switched to using		
\@gls@hyp@opt	146	
\GLSsymbolplural: switched to		
using \@gls@hyp@opt	148	
\Glsymbolplural: switched to		
using \@gls@hyp@opt	148	
\glssymbolplural: switched to		
using \@gls@hyp@opt	147	
\GLStext: switched to using		
\@gls@hyp@opt	140	
\Glstext: switched to using		
\@gls@hyp@opt	140	
\glstext: switched to using		
\@gls@hyp@opt	140	
\glstreenamefmt: new	313	
\GLSuseri: switched to using		
\@gls@hyp@opt	149	
\Glsuseri: switched to using		
\@gls@hyp@opt	149	
\glsuseri: switched to using		
\@gls@hyp@opt	148	
\GLSuserii: switched to using		
\@gls@hyp@opt	150	
\Glsuserii: switched to using		
\@gls@hyp@opt	150	
\glssuserii: switched to using		
\@gls@hyp@opt	149	
\GLSuseriii: switched to using		
\@gls@hyp@opt	151	
\Glsuseriii: switched to using		
\@gls@hyp@opt	151	

\glsuseriii: switched to using \@gls@hyp@opt	150	\@gls@glossary: Renamed \@glossary to \@gls@glossary	199
\GLSuseriv: switched to using \@gls@hyp@opt	152	alttree: moved space after symbol	320, 321
\Glsuseriv: switched to using \@gls@hyp@opt	152	\glspercentchar: new	176
\glsuseriv: switched to using \@gls@hyp@opt	151	\glstildechar: new	176
\GLSuseriv: switched to using \@gls@hyp@opt	153	4.11 – 2014-09-01 \@do@esc@wrglossary: added hook	205
\Glsuseriv: switched to using \@gls@hyp@opt	153	\gls@wrglossary: renamed from \wrglossary to \gls@wrglossary	200
\glsuseriv: switched to using \@gls@hyp@opt	152	\glsaddprotectedpagefmt: new	201
\GLSuservi: switched to using \@gls@hyp@opt	154	\glsbackslash: new	176
\Glsuservi: switched to using \@gls@hyp@opt	154	sanitize: none option	27
\glsuservi: switched to using \@gls@hyp@opt	153	4.12 – 2014-11-22 \@gls@addpredefinedattributes: Added glsignore attribute ..	55
\ifignoredglossary: new	73	\@gls@adjustmode: new	175
\newglossary: added starred version	70	\@gls@notranslatorhook: removed	27
\newignoredglossary: new ...	72	\@gls@toc: added \protect to \numberline	52
\ns@newglossary: added \@gls@type@{name}@log	70	\@gls@usetranslator: new ...	27
new	70	\glsacrpluralsuffix: new ...	43
\p@gls@hyp@opt: new	119	\glsadd: added check for vertical mode	174
\PGLS: changed to use \@gls@hyp@opt	330	\glsaddallunused: replaced @gobble with glsignore ...	175
\Pgls: changed to use \@gls@hyp@opt	328	\glsifusedtranslatordict: new	27
\pgls: changed to use \@gls@hyp@opt	327	\glsignore: new	175
\PGLSpl: changed to use \@gls@hyp@opt	330	\glsupacrpluralsuffix: new .	43
\Pglspl: changed to use \@gls@hyp@opt	329	\ProvidesGlossariesLang: new	43
\pglspl: changed to use \@gls@hyp@opt	328	\RequireGlossariesLang: new .	43
\s@gls@hyp@opt: new	119	4.13 – 2015-02-03 \indexspace: new ...	273, 294, 313
\s@newglossary: new	70	4.14 – 2015-02-28 \@gls@localreset: new	104
4.09 – 2014-08-12 \glsaddkey: fixed bug in user commands	87	\@gls@localunset: new	104
4.10 – 2014-08-27 \@Gls@acrentryname: new	164	\@gls@reset: new	105
\@Gls@entryname: new	164	\@gls@unset: new	104
		\@@newglossaryentry@defcounters: new	106
		\@cGls: new	109
		\@cGls@: new	109
		\@cGlspl@: new	110
		\@cgls: new	109
		\@cgls@: new	109
		\@cglspl: new	110

\@cglspl@: new	110	\@GLSpl: added	
\@gls@entry@count: new	109	\glspostlinkhook	138
\@gls@increment@currcount:		\@Gls@: added	
new	108	\glspostlinkhook	135
\@gls@local@increment@currcount:		\@GLspl@: added	
new	108	\glspostlinkhook	137
\@gls@write@entrycounts: new	108	\@acrlong: added	
\@glslocalreset: new	104	\glspostlinkhook	351
\@glslocalunset: new	104	\@acrshort: added	
\@glsreset: new	104	\glspostlinkhook	350
\@glsunset: new	104	\@gls@: added	
\@newglossaryentry@defcounters:		\glspostlinkhook	134
new	99	\@gls@@link: added	
\cGls: new	109	\glspostlinkhook	120
\cgls: new	109	\@gls@field@link: added	
\cGlsformat: new	110	\glspostlinkhook	139
\cglsformat: new	109	\@gls@link: moved definition of	
\cGlspl: new	110	\glsifhyperon outside of this	
\cglspl: new	110	macro	122
\cGlsplformat: new	111	\@glsdisp: added	
\cglsplformat: new	110	\glspostlinkhook	139
\gls@defdocnewglossaryentry:		\@GLspl@: added	
new	81	\glspostlinkhook	137
\glsenableentrycount: new	106	\glsacspace: new	247
\glslocalreset: switched to		\glsadd: changed	
\@glslocalreset	103	\@do@wrglossary to	
\glslocalunset: switched to		\@do@wrglossary	174
\@glslocalunset	104	\glsfielddef: new	90
\glsreset: switched to		\glsfieldedef: new	89
\@glsreset	103	\glsfieldfetch: new	90
\glsunset: switched to		\glsfieldgdef: new	89
\@glsunset	103	\glsfieldxdef: new	89
4.15 – 2015-03-16		\glsifhyperon: moved definition	
General: bug fix replaced		of \glsifhyperon	121
\@glo@type with \glstype	161	\glslinkpostsetkeys: new	121
4.16 – 2015-06-18		\glspostlinkhook: new	120
\glsaddstoragekey: new	85	\glswriteentry: new	200
4.16 – 2015-07-08		\ifglsfieldcseq: new	92
General: added		\ifglsfielddefeq: new	91
\glspostlinkhook	155–162	\ifglsfieldefeq: new	91
\@ACRlong: added		long-sp-short: new	247
\glspostlinkhook	352	long-sp-short-desc: new	249
\@ACRshort: added		\showglofield: new	266
\glspostlinkhook	351	4.18 – 2015-09-09	
\@Acrlong: added		General: split mfirstuc into	
\glspostlinkhook	352	separate bundle	2
\@Acrshort: added		4.19 – 2015-10-31	
\glspostlinkhook	351	\glstreenamebox: new	319
\@GLS@: added			
\glspostlinkhook	136		

4.19 – 2015-11-22			
\@gls@link@nocheckfirsthyper:		\glsentrytitlecase: new	168
new	139	\glsfindwidesttoplevelname:	
\@gls@preglossaryhook: new	211	new	319
\@printglossary: added		\glslistgroupheaderfmt: new	273
\@gls@preglossaryhook	213	\glslistnavigationitem: new	273
\do@glsdisablehyperinlist:		\glstreegroupheaderfmt: new	313
new	121	\glstreenavigationfmt: new	313
\doifglossarynoexistsordo:		\ifglswrallowprimitivemods:	
new	64	new	203
\gls@gobbleopt: new	70	list: fixed missing space before	
\glsdoifexistsordo: new	64	description	274
4.20 – 2015-11-30		long: fixed typo in	
\@gls@link: added		\glossentrydesc	278
\@gls@setdefault@glslink@opts		mcolaltnreespannav: new	299
.....	122	mcolindexspannav: new	295
added \glsdonohyperlink		mcoltreenonamespannav: new	297
when hyperlink is suppressed	122	mcoltreespannav: new	296
\@gls@setdefault@glslink@opts:		super4col: fixed bug in	
new	121	\glossentry	304
\gls@checkseeallowed@preambleonly:		4.23 – 2016-04-30	
new	76	altlongragged4col: check for	
\glsdonohyperlink: new	132	nogroupskip changed	292
4.21 – 2016-01-24		altsuperragged4col: check for	
General: changed checkfirsthyper		nogroupskip changed	311
assignment	155–162	\glscurrentfieldvalue: new	68
\@printglossary: warn if no		\ifglshasfield: added	
style has been set	211	\glscurrentfieldvalue	67, 68
altlong4col-booktabs: new	285	long: check for nogroupskip	
altlongragged4col-booktabs:		changed	278
new	286	long-booktabs: check for	
\glsLTpenaltycheck: new	287	nogroupskip changed	284
\glspatchLToutput: new	287	long3col: check for nogroupskip	
\glspenaltygroupskip: new	287	changed	279
long-booktabs: new	284	long3col-booktabs: check for	
long3col-booktabs: new	284	nogroupskip changed	284
long4col-booktabs: new	285	long4col: check for nogroupskip	
longragged-booktabs: new	286	changed	281
longragged3col-booktabs: new	286	long4col-booktabs: check for	
\setglossarystyle: set default		nogroupskip changed	285
style if not already set	233	longragged: check for	
4.22 – 2016-04-19		nogroupskip changed	289
\@@do@esc@wrglossary: added		longragged3col: check for	
check for \@arabic	204	nogroupskip changed	291
added test to allow temporary		super: check for nogroupskip	
primitive modifications and		changed	300
added arabic case	204	super3col: check for nogroupskip	
\gls@arabicpage: new	201	changed	302
\gls@protected@pagefmts:		super4col: check for nogroupskip	
added arabic to list	200	changed	304

superragged: check for nogroupskip changed	308	outside of theglossary environment	220
superragged3col: check for nogroupskip changed	310	\glsnavhyperlinkname: new	268
4.24 – 2016-05-27		4.30 – 2017-06-11	
\@gls@extramakeindexopts: new	186	\@glo@autosee: new	98
\@gls@glossary: added check for debug mode	199	\@glo@autoseehook: new	99
\@gls@see@noindex: new	7	\@glo@check@sortallowed: new	14
debug: new	5	\@gls@noidx@do: letter group assignment made global	222
\glsnomakeindexwarning: new	52	\@gls@setupsort@def: added check for register	15
\GlsSetQuote: new	183	\@gls@setupsort@none: new	16
\GlsSetWriteIstHook: new	183	\@xdycrossrefhook: new	57
seenoindex: new	7	\@xdylocationclassorder: bug fix: changed \edef to \def	58
4.25 – 2016-06-09		\glosortentrieswarning: new	21
General: changed		\@gls@set@xr@key: new	75
\DeclareRobustCommand to \newrobustcmd and changed \@ifundefined to \ifcsundef	344	\@gls@xr@key: new	75
\@gls@enablesavenonumberlist: new	77	\GlsAddXdyLocation: bug fix: changed #1 to #2	58
\@gls@initnonumberlist: new	76, 77	\glsnoidxstripaccents: added \a	24
\@gls@savenonumberlist: new	76	added \TH, \dh and \DH	25
4.26 – 2016-10-12		4.31 – 2017-08-10	
\@glossary@default@style: added check for classicthesis	9	nolist: added check for “list” style	11
\glstreechildpredesc: new	314	4.31 – 2017-09-10	
\glstreeitem: new	314	style: changed \renewcommand to \def	9
\glstreepredesc: new	314	4.32 – 2017-08-24	
\glstreesubitem: new	314	\@glsnavhypertarget: new	268
\glstreesubsubitem: new	314	\@glsshowtarget: new	6
mcolindex: replaced \@idxitem with \glstreeitem	294	\glsshowtarget: new	5
mcolindexspannav: replaced \@idxitem with \glstreeitem	295	4.33 – 2017-09-20	
4.28 – 2017-01-07		\@do@esc@wrglossary: added \@gls@the and \@gls@number	205
\glspatchtabularx: new	103	renamed from \@do@esc@wrglossary	203
4.29 – 2017-01-19		\@do@noesc@wrglossary: new	202
General: added check for \@glxtr@doaccsupp	332	\@do@wrglossary: changed to check for esclocations	202
\@gls@noidx@do: current letter group assignment made global	222	\@gls@missinglang@warn: new	21
\@print@noidx@glossary: moved definition of \@gls@currentlettergroup		esclocations: new	10
		\GlsSetXdyFirstLetterAfterDigits: added starred version	176
		\GlsSetXdyNumberGroupOrder: new	177

4.34 – 2017-11-03	
\glsshowtarget: modified to check for math mode and inner	5
mcolalttreespannav: removed spurious space	299
4.35 – 2017-11-14	
\glsadd: added \@gls@setsort (in case of sort=use)	174
4.36 – 2018-03-07	
\@gls@glossary: removed \index	199
4.37 – 2018-04-07	
\gls@begindocdefs: new	83
4.38 – 2018-05-10	
\@gls@define@glossaryentrycounter: added check for existence of glossaryentry counter	12
new	12
\@gls@define@glossarysubentrycounter: new	13
prepended \currentglossary. to \theHglossarysubentry and removed spurious eol space	13
\glsaccsupp: added braces around actual text argument	341
\glsentrycounterlabel: bug fix: move conditional inside command	228
\GlsEntryCounterLabelPrefix: new	225
\glsentryitem: bug fix: move conditional inside command	228
\glsrefentry: bug fix: move conditional inside command	227
\glsresetentrycounter: bug fix: move conditional inside command	227
\glsresetsubentrycounter: bug fix: move conditional inside command	227
\glsstepentry: bug fix: move conditional inside command	227
\glsstepsubentry: bug fix: move conditional inside command	227
\glssubentrycounterlabel: bug fix: move conditional inside command	228
\glssubentryitem: bug fix: move conditional inside command	228
\showglongnameaccess: bug fix: corrected field (was showing text access field)	369
4.40 – 2018-06-01	
\istfile: changed \def to \providecommand	197
\makenoidxglossaries: false . .	194
4.41 – 2018-07-23	
General: changed \val and \nr to \gls@numberedsection@val and \gls@numberedsection@nr	224
\@gls@override@glossary: new	37
debug: changed \val and \nr to \gls@debug@val and \gls@debug@nr	5
\glossary: added warning	37
\gls@original@glossary: new . .	36
\gls@original@makeglossary: new	36
kernelglossredefs: new	37
\makeglossaries: removed redefinition of \makeglossary	192
\makeglossary: added warning	36
nonumberlist: changed \val and \nr to \gls@nonumberlist@val and \gls@nonumberlist@nr	76
numberedsection: changed \val and \nr to \gls@numberedsection@val and \gls@numberedsection@nr . . .	8
seenoindex: changed \val and \nr to \gls@seenoindex@val and \gls@seenoindex@nr . . .	7
translate: changed \val and \nr to \gls@translate@val and \gls@translate@nr	28
4.42 – 2019-01-06	
\@gls@@automake@immediate: new	190
\@gls@automake@immediate: new	189
altlong4col-booktabs: removed superfluous \glspatchLToutput	285

automake: switch from boolean to choice	34	\@GLSuserv@: added accessibility support	357
\gls@automake@nr: new	34	\@GLSuservi@: added accessibility support	357
\glsfielddedef: changed from \edef to \protected@csedef	89	\@GLSdesc@: added accessibility support	354
\glsfieldxdef: changed from \edef to \protected@csxdef	89	\@GLSdescplural@: added accessibility support	354
\ifglsautomake: now defined explicitly instead of through boolean key	34	\@GLSfirst@: added accessibility support	353
noglossaryindex: new	39	\@GLSfirstplural@: added accessibility support	353
4.43 – 2019-09-28		\@GLSname@: added accessibility support	354
\glsnoidxstripaccents: add check for LaTeX version 2019/10/01	25	\@GLSplural@: added accessibility support	353
4.44 – 2019-12-06		\@GLssymbol@: added accessibility support	355
\glsprefix@record@hook: new	327	\@GLssymbolplural@: added accessibility support	355
4.45 – 2020-02-13		\@GLStext@: added accessibility support	352
General: removed backward compatibility use of symbol key	335	\@GLSuseri@: added accessibility support	355
\@do@write@glslabels: new ..	33	\@GLSuserii@: added accessibility support	356
\@glsshowtarget: new	6	\@GLSuseriii@: added accessibility support	356
\@GLSdesc@: added accessibility support	354	\@GLSuseriv@: added accessibility support	356
\@GLSdescplural@: added accessibility support	354	\@GLSuserv@: added accessibility support	357
\@GLSfirst@: added accessibility support	353	\@GLSuservi@: added accessibility support	357
\@GLSfirstplural@: added accessibility support	353	\@do@write@glslabels: new ..	33
\@GLSname@: added accessibility support	354	\@domakeglossaries: new	32
\@GLSplural@: added accessibility support	353	\@gls@access@display: deprecated	341
\@GLSsymbol@: added accessibility support	355	\@gls@fieldaccess@display: new	341
\@GLSsymbolplural@: added accessibility support	355	\@glsdesc@: added accessibility support	354
\@GLStext@: added accessibility support	352	\@glsdescplural@: added accessibility support	354
\@GLSuseri@: added accessibility support	355	\@glsfirst@: added accessibility support	353
\@GLSuserii@: added accessibility support	356	\@glsfirstplural@: added accessibility support	353
\@GLSuseriii@: added accessibility support	356	\@glsname@: added accessibility support	354
\@GLSuseriv@: added accessibility support	356		

\@glsplural@: added accessibility support	353	\Glsentryprefixfirstplural: added \glsdetoklabel	326
\@glsshowaccsupp: new	7	\glentryprefixfirstplural: added \glsdetoklabel	325
\@glssymbol@: added accessibility support	355	\Glsentryprefixplural: added \glsdetoklabel	326
\@glssymbolplural@: added accessibility support	355	\glentryprefixplural: added \glsdetoklabel	325
\@glstext@: added accessibility support	352	\glentrytitlecase: added existence check	168
\@glseri@: added accessibility support	355	\glentryuseriaccess: new	339
\@glserii@: added accessibility support	356	\glentryuseriiaccess: new	339
\@glseriii@: added accessibility support	356	\glentryuseriiiaccess: new	339
\@glseriv@: added accessibility support	356	\glentryuserivaccess: new	339
\@glserv@: added accessibility support	357	\glentryuserviaccess: new	339
\@glservi@: added accessibility support	357	\glsfieldaccsupp: new	340
\@org@Gls@acentryname: added check for \glsshortaccessdisplay	165	\glsfirstaccessdisplay: added check for existence	342
acronyms: changed \renewcommand to \def	18	\glsfirstpluralaccessdisplay: added check for existence	342
debug: showaccsupp	5	\glslongaccessdisplay: added check for existence	343
disablemakegloss: new	32	\glslongpluralaccessdisplay: added check for existence	343
\gls@accessibility: new	332	\glsnameaccessdisplay: added check for existence	341
\gls@accsupp@engine: new	332	\glspluralaccessdisplay: added check for existence	342
\glsaccessibility: new	332	\glsprefixsep: new	327
\glsdefaultshortaccess: new	368	\glssee: switched to \newrobustcmd	208
\glsdescriptionaccessdisplay: added check for existence	342	\glsseeformat: switched to \newrobustcmd	208
\glsdescriptionpluralaccessdisplay: added check for existence	343	\glsseeitem: switched to \newrobustcmd	209
\glentrydescpluralaccess: corrected field reference	338	\glsseelist: switched to \newrobustcmd	208
\glentryfirstpluralaccess: switched to using \@gls@entry@field	338	\glsshortaccessdisplay: added check for existence	343
\glentryparent: new	169	\glsshortaccsupp: new	341
\Glsentryprefix: added \glsdetoklabel	326	\glsshortplaccsupp: new	341
\glentryprefix: added \glsdetoklabel	325	\glsshortpluralaccessdisplay: added check for existence	343
\Glsentryprefixfirst: added \glsdetoklabel	325	\glsshowaccsupp: new	7
\glentryprefixfirst: added \glsdetoklabel	325	\glsshowtargetfont: new	6
		\glsshowtargetouter: new	6
		\glsshowtargetsymbols: new	6
		\glssymbolaccessdisplay: added check for existence	342

<code>\glssymbolpluralaccessdisplay:</code>	<code>\ifglossaryexists:</code> added
added check for existence . 342	starred form 62
<code>\glstextaccessdisplay:</code> added	<code>\s@ifglossaryexists:</code> new . . . 62
check for existence 342	<code>\setglossarypreamble:</code> switched
<code>\glsuseriaccessdisplay:</code> new . 343	to starred form of
<code>\glsuseriiaccessdisplay:</code> new 343	<code>\ifglossaryexists</code> 48
<code>\glsuseriiiaccessdisplay:</code>	4.47 – 2021-09-20
new 344	General: changed <code>\x</code> to <code>\@gls@x</code> 236
<code>\glsuserivaccessdisplay:</code> new 344	<code>\@do@esc@wrglossary:</code> changed
<code>\glsuservaccessdisplay:</code> new . 344	<code>\edef</code> to <code>\protected@edef</code> 205
<code>\glsuserviiaccessdisplay:</code> new 344	<code>\@do@noesc@wrglossary:</code>
<code>\ifglshaschildren:</code> made robust 65	changed <code>\edef</code> to
<code>\ifglshasfield:</code> made robust . 67	<code>\protected@edef</code> 203
<code>\ifglshaslong:</code> made robust . . 66	<code>\@do@write@glslabels:</code> fully
<code>\ifglshasprefix:</code> added	expand list 33
<code>\glsdetoklabel</code> 326	<code>\@gls@@default@acr@descplural:</code>
<code>\ifglshasprefixfirst:</code> added	new 258
<code>\glsdetoklabel</code> 326	<code>\@gls@@default@acr@firstplural:</code>
<code>\ifglshasprefixfirstplural:</code>	new 258
added <code>\glsdetoklabel</code> . . . 327	<code>\@gls@@default@acr@plural:</code>
<code>\ifglshasprefixplural:</code> added	new 258
<code>\glsdetoklabel</code> 326	<code>\@gls@noexpand@assign@default@plurals:</code>
<code>\ifglshasshort:</code> made robust . 66	new 258
<code>\ifglshassymbol:</code> made robust 66	<code>\@glo@do@sortentries:</code> changed
<code>\makeglossaries:</code> let	<code>\edef</code> to <code>\protected@edef</code> 216
<code>\@makeglossary</code> to <code>\@gobble</code>	<code>\@glo@sortentries:</code> changed
instead of <code>\relax</code> 192	<code>\edef</code> to <code>\protected@edef</code> 215
<code>restoremakeloss:</code> new 32	<code>\@glo@storeentry:</code> changed
<code>user1access:</code> new 334	<code>\edef</code> to <code>\protected@edef</code> 100
<code>user2access:</code> new 334	<code>\@gls@checkesc:</code> new 130
<code>user3access:</code> new 334	<code>\@gls@deprecated@acr@warning:</code>
<code>user4access:</code> new 335	new 260
<code>user5access:</code> new 335	<code>\@gls@link:</code> changed <code>\edef</code> to
<code>user6access:</code> new 335	<code>\protected@edef</code> 121
<code>writeglslabels:</code> new 34	<code>\@gls@setupsort@use:</code> changed
<code>\xglslfieldaccsupp:</code> new 340	<code>\edef</code> to <code>\protected@edef</code> . 16
4.46 – 2020-03-19	<code>\@gls@writedef:</code> added a
<code>\@ifglossaryexists:</code> new 62	warning if entry has already
<code>\@printgloss@checkexists:</code>	been defined 84
new 210	<code>\@makeglossaries@warn@noprintglossary:</code>
<code>\@printgloss@checkexists@allowignored:</code>	new 193
new 210	<code>\@write@glslabelsname:</code> new . 33
<code>\@printgloss@checkexists@noignored:</code>	<code>\do@write@glslabelsname:</code> new 33
new 210	<code>\forglsentries:</code> changed <code>\edef</code>
<code>\@printglossary:</code> replaced	to <code>\protected@edef</code> 62
<code>\ifglossaryexists</code> with	<code>\gls@defdocnewglossaryentry:</code>
<code>\@printgloss@checkexists</code> 212	removed see warning 81
<code>\doifglossarynoexistsordo:</code>	<code>\gls@defglossaryentry:</code> changed
switched to starred form of	<code>\edef</code> to
<code>\ifglossaryexists</code> 64	<code>\protected@edef</code> 93, 94

<code>\gls@docdef@addlabel: new</code>	82	<code>\SetDescriptionFootnoteAcronymStyle:</code>	
<code>\gls@docdef@labellist: new</code>	82	added deprecated warnings	259
<code>\gls@glsdefs@addlabel: new</code>	82	<code>\SetDUASStyle: added deprecated</code>	
<code>\gls@glsdefs@labellist: new</code>	82	warnings	260
<code>\gls@save@numberlist: changed</code>		<code>\SetFootnoteAcronymStyle:</code>	
<code>\edef to \protected@edef</code>	209	added deprecated warnings	259
<code>\glsadd: changed \edef to</code>		<code>\SetSmallAcronymStyle: added</code>	
<code>\protected@edef</code>	174	deprecated warnings	260
<code>\glsaddall: changed \edef to</code>		<code>writeglslabelnames: new</code>	34
<code>\protected@edef</code>	175	4.48 – 2021-10-19	
<code>\glsdefs@newdocentry: new</code>	83	<code>\glscapitalisewords: new</code>	168
<code>\glsdisplaynumberlist: changed</code>		<code>\glslistexpandedname: new</code>	274
<code>\edef to \protected@edef</code>	173	<code>\glslistinit: new</code>	273
<code>\glsfieldddef: changed \edef to</code>		<code>\glsunexpandedfieldvalue: new</code>	90
<code>\protected@edef</code>	90	4.49 – 2021-11-01	
<code>\glsfielddedef: changed \edef to</code>		<code>\glslistinit: added</code>	
<code>\protected@edef</code>	89	<code>\glsnavigation</code>	273
<code>\glsfieldfetch: changed \edef</code>		4.50 – 2022-10-14	
<code>to \protected@edef</code>	90	General: added initialisation	238
<code>\glsfieldgdef: changed \edef to</code>		allow markup to be stripped	
<code>\protected@edef</code>	89	from location	236
<code>\glsfieldxdef: changed \edef to</code>		corrected description mapping	335
<code>\protected@edef</code>	89	don't load glossary-super if xtab	
<code>\glsIfListOfAcronyms: changed</code>		has been loaded	10
<code>\edef to \protected@edef</code>	19	No longer requires textcase	
<code>\glsmoveentry: changed \edef to</code>		with mfirstuc 2.08+	3
<code>\protected@edef</code>	99	removed automatic loading of	
<code>\glsnavhyperlink: changed</code>		glossaries-compatible-307	40
<code>\edef to \protected@edef</code>	268	switched to using	
<code>\GlsSetQuote: added check for</code>		<code>\glsuppercase</code>	
<code>backslash</code>	185	140–154, 156, 158, 160, 162
<code>\glsstepsubentry: changed</code>		<code>\@do@esc@wrglossary: moved</code>	
<code>\edef to \protected@edef</code>	227	<code>\romannumeral</code> inside allow	
<code>\ifglsfieldcseq: changed \edef</code>		primitive mods	205
<code>to \protected@edef</code>	92	<code>\@gls@default@acr@descplural:</code>	
<code>\ifglshaschildren: changed</code>		removed	258
<code>\edef to \protected@edef</code>	65	<code>\@gls@default@acr@firstplural:</code>	
<code>\ifignoredglossary: changed</code>		removed	258
<code>\edef to \protected@edef</code>	73	<code>\@gls@default@acr@plural:</code>	
<code>\makenoidxglossaries: changed</code>		removed	258
<code>\edef to \protected@edef</code>	194	<code>\@gls@default@entryfmt:</code>	
<code>\new@glossaryentry: allow</code>		removed	344
<code>overwrite if defined in</code>		<code>\@gls@navhypertarget: new</code>	268
<code>.glsdefs</code>	82	<code>\@gls@noexpand@assign@default@plurals:</code>	
<code>\newignoredglossary: changed</code>		removed	258
<code>\edef to \protected@edef</code>	72	<code>\Gls@acentryname: removed</code>	
<code>\ns@newglossary: changed \edef</code>		mfirstuc internal commands	164
<code>to \protected@edef</code>	70	<code>\@Glsdisp: new</code>	139
<code>\SetDescriptionDUAAcronymStyle:</code>		<code>\@Glsentryfield: new</code>	163
added deprecated warning	259		

\@domakeglossaries: added		
check for \nofiles	32	
\@gls@automake: added check for		
void codepage	188	
\@gls@automake@immediate:		
added check for void		
codepage	189	
\@gls@automake@types: new	190	
\@gls@defaultttype: new	69	
\@gls@do@automake@err: new	34	
\@gls@do@glsunset: new	118	
\@gls@entry@field: switched to		
\csuse	162	
\@gls@escbsdq: check		
esclocations setting	125	
\@gls@getcounterprefix:		
changed \edef to \def	207	
\@gls@getgrouptitle: switched		
to L ^A T _E X3 syntax	232	
\@gls@glslink@hyper@update@hook:		
new	118	
\@gls@hypergroup: updated to		
use etoolbox	269	
\@gls@noist@toolate: new	187	
\@gls@removedoption: new	25	
\@gls@requested@languages:		
new	29	
\@gls@restore@glslocal: new	118	
\@gls@sanitizedesc: removed	22	
\@gls@save@glslocal: new	118	
\@gls@setupsort@clear: new	16	
\@gls@defaultplural: Removed	78	
\@org@Gls@acrentryname: saved		
original definition of		
\@Gls@acrentryname	165	
\@xp@gls@getcounterprefix:		
new	206	
\@xp@gls@hypernumber: new	235	
automake: added makegloss and		
lite options	34	
automakegloss: new	35	
automakeglosslite: new	35	
compatible-2.07: deprecated	36	
compatible-3.07: deprecated	36	
\compatibleglossentry:		
removed	229, 332	
\compatiblesubglossentry:		
removed	230, 332	
\CustomAcronymFields:		
removed	261	
\CustomNewAcronymDef:		
removed	261	
\DefaultNewAcronymDef:		
removed	258, 368	
\defglsdisplay: removed	117	
\defglsdisplayfirst: removed	117	
\DescriptionDUANewAcronymDef:		
removed	259	
\DescriptionFootnoteNewAcronymDef:		
removed	258, 368	
\DescriptionNewAcronymDef:		
removed	259, 368	
dua: switched to using		
\glsupercase	251, 252	
dua-desc: removed \acronymfont		
from around long form	253	
\DUANewAcronymDef: removed	260	
esclocations: default now false	10	
\findrootlanguage: removed	60	
footnote: switched to using		
\glsupercase	254	
\FootnoteNewAcronymDef:		
removed	259, 368	
\glossary: removed	37	
\glossaryentryfield: removed	230	
\glossarymark: switched to using		
\glsupercase	50	
\glossarystyle: removed	233	
\glossarysubentryfield:		
removed	231	
\gls@automake@makegloss: new	34	
\gls@automake@option: new	34	
\gls@codepage: set default		
codepage to utf8 if		
\inputencodingname not		
defined and switched to		
\ifdefvoid	31	
\gls@delimN: renamed \@delimN		
to \gls@delimN	236	
\gls@delimR: renamed \@delimR		
to \gls@delimR	235	
\gls@disablepagerefexpansion:		
added check for esclocations	200	
\gls@noidx@nonumberlist: new	76	
\gls@noidxglossary: added		
\glswrglossdisablelocationcmds		
	197	
\gls@xdy@set@lang: new	31	
\glsaddkey: allow expansion in		
PDF bookmarks	87	

switched to using		
\glssupercase	88	
\GlsAddXdyAttribute: remove		
duplicate counterless		
attribute definitions	54	
\Glsdisp: new	139	
\glsdisplay: removed	117	
\glsdisplayfirst: removed	117	
\glsdoparenifnotempty:		
replaced		
\DeclareRobustCommand with		
\newrobustcmd	259	
\glsdoshowtarget: new	6	
\glsencapwrcontent: new	122	
\Glsentrydesc: allow expansion		
in PDF bookmarks	166	
\Glsentrydescplural: allow		
expansion in PDF		
bookmarks	166	
\Glsentryfirst: allow expansion		
in PDF bookmarks	167	
\Glsentryfirstplural: allow		
expansion in PDF		
bookmarks	168	
\glsentryfmt: obsolete		
commands replaced with		
\glsgenentryfmt	112	
\Glsentrylong: allow expansion		
in PDF bookmarks	171	
\Glsentrylongpl: allow		
expansion in PDF		
bookmarks	171	
\Glsentryname: allow expansion		
in PDF bookmarks	164	
\Glsentryplural: allow		
expansion in PDF		
bookmarks	167	
\Glsentryprefix: allow		
expansion in PDF		
bookmarks	326	
\Glsentryprefixfirst: allow		
expansion in PDF		
bookmarks	325	
\Glsentryprefixfirstplural:		
allow expansion in PDF		
bookmarks	326	
\Glsentryprefixplural: allow		
expansion in PDF		
bookmarks	326	
\Glsentryshort: allow expansion		
in PDF bookmarks	171	
\Glsentryshortpl: allow		
expansion in PDF		
bookmarks	171	
\Glsentrysymbol: allow		
expansion in PDF		
bookmarks	167	
\Glsentrysymbolplural: allow		
expansion in PDF		
bookmarks	167	
\Glsentrytext: allow expansion		
in PDF bookmarks	166	
\glsentrytitlecase: use		
sentence case for PDF		
bookmark	168	
\Glsentryuseri: allow expansion		
in PDF bookmarks	169	
\Glsentryuserii: allow		
expansion in PDF		
bookmarks	169	
\Glsentryuseriii: allow		
expansion in PDF		
bookmarks	169	
\Glsentryuseriv: allow		
expansion in PDF		
bookmarks	170	
\Glsentryuserv: allow expansion		
in PDF bookmarks	170	
\Glsentryuservi: allow		
expansion in PDF		
bookmarks	170	
\glsfielddef: made robust	90	
\glsfieldedef: made robust	89	
\glsfieldfetch: made robust	90	
\glsfieldgdef: made robust	89	
\glsfieldxdef: made robust	89	
\glsgenacfmt: switched to using		
\glssupercase	114	
\glsgenentryfmt: switched to		
using \glssupercase	113	
\glsgetgrouplabel: converted to		
use L ^A T _E X3	232	
\glsgetgrouptitle: made		
robust	232	
\gls glossarymark: switched to		
\glssupercase	49	
\glsifhyper: removed	119	
\glsindexingsetting: new	13	
\glsinlineifhaschildren: new	272	

\Glslink: new	120	\SetCustomDisplayStyle:	
\glslocationcstoencap: new	48	removed	261
\glslowercase: new	2	\SetCustomStyle: removed	261
\glsmfuaddmap: new	3	\SetDefaultAcronymStyle:	
\glsmfublocker: new	3	removed	258
\glsmfuexcl: new	3	\SetDescriptionAcronymDisplayStyle:	
\glsnavhyperlink: changed		removed	259
\@glo@type to		\SetDescriptionAcronymStyle:	
\currentglossary	268	removed	259
\glsnavhypertarget: changed		\SetDescriptionDUAAcronymDisplayStyle:	
\@glo@type to		removed	259
\currentglossary	268	\SetDescriptionDUAAcronymStyle:	
\glsnavigation: changed		removed	259
\@glo@type to		\SetDescriptionFootnoteAcronymDisplayStyle:	
\currentglossary	269	removed	258
switched to \letcs	270	\SetDescriptionFootnoteAcronymStyle:	
\glsnoidxprenumberlist: new	76	removed	259
\glsnumberlistloop: corrected		\SetDUADisplayStyle: removed	260
parameter number	197	\SetDUASStyle: removed	260
\glsseeitemformat: added test		\SetFootnoteAcronymDisplayStyle:	
for short field	209	removed	259
\glsentencecase: new	2	\SetFootnoteAcronymStyle:	
\glsshowtargetfonttext: new	6	removed	259
\glsshowtargetinner: new	6	\SetGenericNewAcronym:	
\glstexorpdfstring: new	163	corrected long plural suffix	244
\glssupercase: new	2	switched to using	
\GlsWarnAddProtectedPageFmt:		\glssupercase	244
new	202	\setglossentrycompatibility:	
\glswrglossdisableanchorcmds:		removed	230
new	202	\SetSmallAcronymDisplayStyle:	
\glswrglossdisablelocationcmds:		removed	260
new	201	\SetSmallAcronymStyle:	
\glswrglosslocationtarget:		removed	260
new	202	\SmallNewAcronymDef:	
\glswrglosslocationtextfmt:		removed	260, 368
new	202	sort: added clear	13
\ifglsfieldcseq: made robust	92	4.51 – 2022-11-02	
\ifglsfielddefeq: made robust	91	General: switch to	
\ifglsfielddeq: made robust	91	\TrackIfKnownLanguage	43
\ifglsfieldvoid: new	66	\@do@wrglossary: added	
\ifglsresetcurrcount: new	106	\gls@ifnotmeasuring	200
\ifglswrallowprimitivemods:		\@gls@@automake@immediate:	
switch default to false	203	ensure list is expanded before	
\istfile: removed	197	iterating	190
kernelglossredefs: corrected		\@gls@if@measuring: new	101
names of choice variables	37	alttree: replaced \settowidth	
languages: new	29	with \glsmeasurewidth	320, 321
\makeglossary: removed	36	\gls@if@measuring: new	101
mfirstuc: new	39	\gls@local@stepcounter: new	102
\SetAcronymStyle: removed	260	\gls@org@refstepcounter: new	101

<code>\gls@start@measuring:</code> new ..	102	<code>\glsmeasuredepth:</code> new	102
<code>\gls@stop@measuring:</code> new ...	102	<code>\glsmeasureheight:</code> new	102
<code>\glsdohypertarget:</code> added		<code>\glsmeasurewidth:</code> new	103
measuring and group	132	<code>\glspatchtabularx:</code> switch	
<code>\glsfindwidesttoplevelname:</code>		measuring on and off	103
replaced <code>\settowidth</code> with		<code>\makeglossaries:</code> ensure list of	
<code>\glsmeasurewidth</code>	319	glossaries is expanded before	
<code>\glsifmeasuring:</code> new	102	iterating	193

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	
\!	126, 128, 185, 445, 446, 694, 696, 697, 727
\"	24, 125, 127, 128, 131, 382, 443–445, 447, 632, 694–697, 699
\#	130, 131, 180, 470, 471, 698, 699, 722
\%	176, 182, 468, 472, 573–575, 720, 724, 826–828
\&	43, 173, 393, 467, 643, 719
\'	24, 382, 632
\.	11, 24, 375, 382, 625, 632
\=	24, 382, 632
\?	125, 127, 185, 443–445, 474, 694–696, 726
\@	83, 415, 666
\@@delimN	505, 757
\@do@wrglossary	194, 203, 205, 206, 480, 486, 487, 732, 737, 738
\@do@esc@wrglossary	202, 203, 485, 486, 737, 738
\@do@noesc@wrglossary	202, 202, 485, 737
\@do@wrglossary	174, 200, 202, 467, 484, 485, 719, 736, 737
\@do@write@glslabels	33, 33, 388, 638
\@glo@assign@sortkey	195, 226, 481, 499, 733, 751
\@glo@list	62, 404, 654
\@glo@sort	24, 382, 632
\@glo@type	211, 489, 741
\@glossarysec	8, 8, 50, 51, 374, 397, 623, 624, 647
\@glossaryseclabel	8, 8, 9, 50, 51, 224, 225, 374, 397, 499, 624, 647, 750, 751
\@glossarysecstar	8, 8, 9, 50, 51, 224, 225, 374, 397, 499, 624, 647, 750, 751
\@gls@@default@acr@descplural	258, 771
\@gls@@default@acr@firstplural	258, 771
\@gls@@default@acr@plural	258, 771
\@gls@checkactual	130, 447, 698
\@gls@checkbar	129, 446, 697
\@gls@checkescactual	127, 444, 445, 695, 696
\@gls@checkescbar	128, 445, 696
\@gls@checkesclevel	128, 129, 445, 446, 696, 697
\@gls@checkescquote	127, 186, 444, 475, 476, 695, 727, 728
\@gls@checklevel	129, 130, 446, 447, 697, 698
\@gls@checkquote	126, 184, 443, 444, 474, 694, 695, 725, 726
\@gls@codepage	188, 189
\@gls@default@entryfmt	344, 432, 438, 439, 543, 683, 689, 796
\@gls@expand@field	22, 80, 80, 86, 381, 413, 414, 417, 519–522, 524, 526, 528, 530, 564–568, 631, 664, 667, 668, 771, 773, 774, 776, 778, 779, 781, 783, 818–821
\@gls@extramakeindexopts	184, 191, 473, 479, 725, 731
\@gls@fixbraces	208, 208, 488, 740
\@gls@navhypertarget	268, 268
\@gls@noexpand@assign@default@plurals	258, 771, 818
\@gls@noexpand@field	22, 79, 80, 381, 413, 631, 663, 664
\@gls@noidx@no@sanitizesort	24, 382, 632
\@gls@noidx@nosanitizesort	24, 196, 382, 482, 632, 734
\@gls@noidx@sanitizesort	23, 382, 631
\@gls@nosanitizesort	23, 23, 196, 381, 382, 482, 631, 734
\@gls@sanitizesort	23, 23, 196, 381, 382, 482, 631, 734
\@gls@xdycheckbackslash	131, 132, 448, 699, 700
\@gls@xdycheckquote	131, 447, 699

\@glslocalreset	104, 104, 107, 428, 430, 679, 680	\@GLSsymbol@	147, 355, 455, 553, 707, 806
\@glslocalunset	104, 104, 107, 428, 429, 679, 680	\@GLSsymbolplural	148, 455, 707
\@glsreset	104, 105, 107, 428, 429, 679, 680	\@GLSsymbolplural@	148, 355, 455, 553, 707, 807
\@glsshowtarget	5, 6, 7, 372, 373, 622, 623	\@GLStext	140, 452, 704
\@glsunset	104, 104, 107, 428, 429, 679, 680	\@GLStext@	140, 352, 452, 552, 704, 805
\@newglossaryentry@defcounters	106, 106, 429, 680	\@GLSuseri	149, 456, 707
\@this@glo@	62, 404, 654	\@GLSuseri@	149, 355, 456, 554, 707, 807
\@ACRfull	241, 242, 507, 759	\@GLSuserii	150, 456, 708
\@ACRfullpl	242, 507, 508, 759	\@GLSuserii@	150, 356, 456, 554, 708, 807
\@ACRlong	160, 240, 352, 461, 507, 551, 712, 759, 804	\@GLSuseriii	151, 456, 708
\@ACRlongpl	162, 241, 462, 508, 713, 760	\@GLSuseriii@	151, 356, 456, 554, 708, 807
\@ACRshort	156, 240, 351, 458, 507, 550, 710, 759, 804	\@GLSuseriv	152, 457, 708
\@ACRshortpl	158, 241, 459, 460, 508, 711, 760	\@GLSuseriv@	152, 356, 457, 554, 708, 807
\@Acrfull	241, 507, 759	\@GLSuserv	153, 457, 709
\@Acrfullpl	242, 507, 759	\@GLSuserv@	153, 357, 457, 555, 709, 808
\@Acrlong	159, 240, 351, 460, 507, 551, 712, 759, 804	\@GLSuservi	154, 457, 709
\@Acrlongpl	161, 241, 461, 507, 713, 759	\@GLSuservi@	154, 357, 457, 555, 709, 808
\@Acrshort	155, 350, 458, 550, 710, 803	\@Gls	134, 135, 449, 701
\@Acrshortpl	157, 459, 711	\@Gls@	107, 109, 135, 135, 329, 430, 431, 449, 589, 681, 682, 701, 842
\@Alph	201, 204, 205, 485, 486, 737, 738	\@Gls@@link	120
\@GLS	135, 450, 701, 702	\@Gls@acentryname	164, 243, 463, 508, 714, 760
\@GLS@	135, 135, 330, 450, 590, 702, 843	\@Gls@entry@field	163, 163, 164, 165, 418, 462–466, 668, 714–717
\@GLSdesc	145, 454, 706	\@Gls@entryname	164, 164, 243, 463, 508, 714, 760
\@GLSdesc@	145, 354, 454, 553, 706, 806	\@GlsSetXdyFirstLetterAfterDigits	176, 177, 468, 720
\@GLSdescplural	146, 455, 706	\@GlsSetXdyNumberGroupOrder	177, 468, 469, 720
\@GLSdescplural@	146, 354, 455, 553, 706, 806	\@Glsdesc	145, 454, 706
\@GLSfirst	141, 453, 704	\@Glsdesc@	145, 354, 454, 553, 706, 806
\@GLSfirst@	141, 353, 453, 552, 704, 805	\@Glsdescplural	146, 454, 706
\@GLSfirstplural	143, 453, 705	\@Glsdescplural@	146, 354, 454, 553, 706, 806
\@GLSfirstplural@	143, 353, 454, 552, 705, 806	\@Glsdisp	139, 139
\@GLSname	144, 454, 705	\@Glsentryfield	87, 163, 166–171, 325, 326
\@GLSname@	144, 354, 454, 553, 706, 806	\@Glsfirst	141, 453, 704
\@GLSpl	138, 138, 451, 703	\@Glsfirst@	141, 353, 453, 552, 704, 805
\@GLSpl@	138, 331, 451, 591, 703, 844	\@Glsfirstplural	143, 453, 705
\@GLSplural	142, 453, 705	\@Glsfirstplural@	143, 353, 453, 705
\@GLSplural@	142, 353, 453, 552, 705, 805	\@Glsname	144, 454, 705
\@GLSsymbol	147, 455, 707		

<code>\@Glsname@</code>	144, 354 , 454, 552, 705, 806	<code>\@afterheading</code>	275, 577, 594, 830, 848
<code>\@Glspl</code>	137, 451, 702	<code>\@alph</code>	201, 204, 205, 485, 486, 737, 738
<code>\@Glspl@</code>	108,	<code>\@arabic</code>	201, 204, 205, 485, 486, 737, 738
	111, 137, 137 , 330, 430, 432,	<code>\@auxout</code>	69, 71, 108,
	451, 589, 590, 681, 682, 702, 843		191, 192, 194, 197, 209, 214,
<code>\@Glsplural</code>	142, 453, 705		268, 408, 431, 479–481, 483,
<code>\@Glsplural@</code>	142, 353 , 453, 552, 705, 805		489, 491, 591, 658, 659, 681,
<code>\@Glsymbol</code>	147, 455, 706		682, 731–733, 735, 741, 743, 844
<code>\@Glsymbol@</code>	147, 355 , 455, 553, 706, 806	<code>\@backslashchar</code>	125, 126, 131, 132, 185,
<code>\@Glsymbolplural</code>	148, 455, 707		442, 448, 693, 694, 699, 700, 727
<code>\@Glsymbolplural@</code>	148, 355 , 455, 553, 707, 807	<code>\@bsphack</code>	199, 484, 736
<code>\@Glstext</code>	140, 452, 704	<code>\@cGls</code>	109, 109 , 431, 682
<code>\@Glstext@</code>	140, 352 , 452, 552, 704, 805	<code>\@cGls@</code>	107, 109, 109 , 430, 431, 681, 682
<code>\@Glsuser@i</code>	355–357, 554, 555, 807, 808	<code>\@cGlspl</code>	110, 431, 682
<code>\@Glsuseri</code>	149, 456, 707	<code>\@cGlspl@</code>	108, 110, 110 , 430–432, 681, 682
<code>\@Glsuseri@</code>	149, 355 , 456, 707	<code>\@cclv</code>	287, 600, 601, 854
<code>\@Glsuserii</code>	150, 456, 708	<code>\@cgls</code>	109, 109 , 431, 682
<code>\@Glsuserii@</code>	150, 356 , 456, 708	<code>\@cgls@</code>	107, 109, 109 , 430, 431, 680, 682
<code>\@Glsuseriii</code>	151, 456, 708	<code>\@cglspl</code>	110, 110 , 431, 682
<code>\@Glsuseriii@</code>	151, 356 , 456, 708	<code>\@cglspl@</code>	107, 110, 110 , 430, 431, 681, 682
<code>\@Glsuseriv</code>	152, 457, 708	<code>\@chapter</code>	41, 392, 642
<code>\@Glsuseriv@</code>	152, 356 , 457, 708	<code>\@classoptionslist</code>	39, 391, 641
<code>\@Glsuserv</code>	153, 457, 709	<code>\@closegls</code>	188, 188 , 476, 477, 728, 729
<code>\@Glsuserv@</code>	153, 357 , 457, 709	<code>\@colht</code>	287, 601, 854
<code>\@Glsuservi</code>	154, 457, 709	<code>\@colroom</code>	287, 288, 601, 854
<code>\@Glsuservi@</code>	154, 357 , 457, 709	<code>\@currentlabelname</code>	9, 225, 374, 499, 624, 751
<code>\@Mi</code>	287, 600, 854	<code>\@curroptions</code>	39, 391, 641
<code>\@PGLS</code>	330, 330 , 590, 843	<code>\@declaredoptions</code>	39, 391, 641
<code>\@PGLS@</code>	330, 330 , 590, 843	<code>\@delimN</code>	505, 757
<code>\@PGLSpl</code>	330, 331 , 590, 843	<code>\@delimR</code>	505, 757
<code>\@PGLSpl@</code>	331, 331 , 590, 843	<code>\@disable@onlypremakeg</code>	41, 192, 392, 479, 643, 731
<code>\@Pgls</code>	328, 328 , 589, 842	<code>\@disable@premakecs</code>	41, 41 , 393, 643
<code>\@Pgls@</code>	328, 328 , 589, 842	<code>\@disabled@glsaddxdycounters</code>	53, 55, 399, 649, 650
<code>\@Pglspl</code>	329, 329 , 589, 842	<code>\@do@addcounter</code>	53, 398, 399, 649
<code>\@Pglspl@</code>	329, 329 , 589, 842	<code>\@do@auxoutstuff</code>	214, 491, 743
<code>\@Roman</code>	201, 204, 205, 485, 486, 737, 738	<code>\@do@glossentry</code>	501, 535, 753, 788
<code>\@acrfull</code>	241, 241 , 506, 758	<code>\@do@gls@checkesc</code>	126, 694
<code>\@acrfullpl</code>	242, 507, 759	<code>\@do@gls@getcounterprefix</code>	206, 485, 486, 737, 738
<code>\@acrlong</code>	158, 240, 351 ,	<code>\@do@gls@islistofacronyms</code>	19, 379, 629
	460, 506, 551, 711, 712, 758, 804	<code>\@do@glssee</code>	98, 425, 676
<code>\@acrlongpl</code>	160, 240, 461, 507, 713, 759	<code>\@do@ifinlist</code>	53, 398, 648
<code>\@acrshort</code>	155, 240, 350 , 458,		
	506, 507, 550, 709, 758, 759, 803		
<code>\@acrshortpl</code>	156,		
	157, 240, 241, 459, 507, 710, 759		
<code>\@addtoacronymlists</code>	19, 379, 629		
<code>\@addtoacronymlists</code>	19		

<code>\@do@newglossaryentry</code>	<code>\@for</code>
244, 508, 518–531,	29, 39, 41,
564–568, 760, 770–785, 817–821	43, 53, 54, 61, 62, 84, 85, 125,
<code>\@do@seeglossary</code>	178–180, 190, 192, 193, 200,
194,	208, 215, 245, 269, 270, 386,
207, 208, 480, 488, 732, 739, 740	391, 392, 398, 399, 404, 415,
<code>\@do@subglossentry</code> 502, 535, 754, 788	416, 442, 469, 470, 479, 480,
<code>\@do@wrglossary</code>	485, 488, 492, 509, 519, 521,
122,	523, 524, 526, 529, 530, 532,
200, 441, 484, 571, 691, 736, 824	571, 591, 592, 636, 641, 643,
<code>\@do@write@glslabels</code> 33, 34, 388, 638	649, 654, 666, 667, 693, 721,
<code>\@do@writeaux@info</code>	722, 731, 732, 737, 740, 743,
209, 489, 741	744, 761, 771, 774, 775, 777,
<code>\@domakeglossaries</code>	779, 782, 783, 785, 825, 844, 845
. . . 32, 32, 33, 191, 193, 387,	<code>\@glo@desc</code>
388, 479, 480, 637, 638, 731, 732	97, 424, 675
<code>\@ehc</code>	<code>\@glo@symbol</code>
287, 600, 854	97, 425, 676
<code>\@empty</code>	<code>\@glo@access</code>
12,	333, 336, 341, 535,
16, 19, 36, 38, 39, 41, 53, 54,	537, 538, 540, 541, 789–791, 794
58, 60, 61, 95, 100, 123, 134–	<code>\@glo@addchildren</code>
138, 155–162, 178, 181, 183,	215, 215, 219, 492, 495, 744, 747
188–190, 198, 199, 207, 233,	<code>\@glo@assign@sortkey</code>
350–352, 376, 378, 379, 389,	192, 195,
391, 392, 398, 399, 401, 403,	226, 480, 481, 499, 731, 733, 751
404, 423, 426, 441, 449–451,	<code>\@glo@autosee</code>
458–462, 469, 471, 473, 477,	98, 98, 425, 676
478, 483, 484, 487, 503, 519,	<code>\@glo@autoseehook</code>
521–528, 530, 532, 550, 551,	98, 99, 425, 676
571, 573–575, 625, 627, 629,	<code>\@glo@check@mkidxrangechar</code> 124,
639, 641, 642, 648, 649, 652,	206, 442, 487, 571, 693, 738, 824
654, 673, 674, 677, 692, 701–	<code>\@glo@check@sortallowed</code>
703, 709–714, 721, 723, 725,	14, 14, 15–17, 193, 196, 376–
728–730, 735, 736, 739, 755,	378, 480, 482, 626–628, 732, 734
771, 773, 775, 777, 779–781,	<code>\@glo@childlist</code>
783, 785, 803–805, 825, 827, 828	215, 492, 744
<code>\@end@fixbraces</code>	<code>\@glo@counter</code>
208, 488, 740	75,
<code>\@end@fortrue</code>	93, 97, 411, 422, 424, 661, 673, 675
29, 65, 85, 269, 386,	<code>\@glo@counterprefix</code> 197, 202, 205–
406, 416, 591, 636, 656, 667, 844	207, 233, 236, 483, 485–487,
<code>\@esphack</code>	503, 505, 735, 737–739, 755, 757
200, 484, 736	<code>\@glo@default@sorttype</code>
<code>\@expandtwoargs</code> 13, 13, 195, 217–219, 376,
39, 391, 641	481, 494, 495, 626, 733, 745–747
<code>\@firstofone</code>	<code>\@glo@defaultcounter</code> 96, 97, 424, 675
24, 33,	<code>\@glo@desc</code>
274, 382, 383, 388, 632, 638, 847	73, 92, 93, 95, 97,
<code>\@firstofthree</code>	410, 421–425, 660, 672, 673, 675
119, 133,	<code>\@glo@descaccess</code>
134, 136, 139, 155, 157, 159,	334, 336, 337, 536–538, 789–791
161, 350–352, 439, 449, 450,	<code>\@glo@descplural</code>
452, 458–461, 550, 551, 690,	73,
701–703, 709, 711–713, 803–805	92, 93, 410, 421, 422, 660, 672
<code>\@firstoftwo</code>	<code>\@glo@descpluralaccess</code>
27–29, 84, 85, 102, 119, 136–	334, 336, 337, 536–538, 789–791
138, 157, 158, 161, 162, 385,	<code>\@glo@do@sortentries</code>
386, 415, 416, 439, 450, 451,
459–462, 634–636, 666, 667,	215, 215, 492, 743, 744
690, 702, 703, 710, 711, 713, 714	<code>\@glo@entry</code>
	175, 468, 719

<code>\@glo@entryprefix</code>	<code>\@glo@plural</code> .. 74, 93, 95, 96, 410,
.... 324, 325, 586, 587, 839, 840	422–424, 566, 660, 673–675, 819
<code>\@glo@entryprefixfirst</code>	<code>\@glo@pluralaccess</code>
.... 324, 325, 586, 587, 839, 840 333, 336, 536–538, 789–791
<code>\@glo@entryprefixfirstplural</code> ..	<code>\@glo@prefix</code> 10, 76, 94, 100,
.... 324, 325, 586, 587, 839, 840	101, 124, 206, 375, 411, 422,
<code>\@glo@entryprefixplural</code>	426, 427, 442, 487, 571, 624,
.... 324, 325, 586, 587, 839, 840	661, 673, 677, 678, 693, 739, 824
<code>\@glo@esclabel</code>	<code>\@glo@range</code> ... 206, 487, 571, 739, 824
.... 100, 101, 426, 427, 677, 678	<code>\@glo@see</code>
<code>\@glo@etext</code>	93, 98, 411, 422, 425, 661, 673, 676
432–434, 683–685	<code>\@glo@seeautonumberlist</code>
<code>\@glo@first</code> 10, 10, 75, 375, 411, 624, 661
74,	<code>\@glo@short</code>
93, 96, 97, 410, 422, 424, 425,	66, 78, 94, 97, 406, 412, 422,
528, 567, 660, 673, 675, 781, 821	424, 567, 657, 662, 673, 675, 821
<code>\@glo@firstaccess</code>	<code>\@glo@shortaccess</code>
.... 333, 336, 536–538, 789–791 334, 336, 337, 536–538,
<code>\@glo@firstplural</code> 74, 93, 96, 410,	564–567, 789–791, 817–819, 821
422, 424, 567, 660, 673, 675, 821	<code>\@glo@shortpl</code>
<code>\@glo@firstpluralaccess</code>	78,
.... 333, 336, 536–538, 789–791	94, 97, 412, 422, 424, 518, 520,
<code>\@glo@grabfirst</code> .. 221, <u>221</u> , 496, 748	522, 524, 525, 528, 529, 564,
<code>\@glo@label</code> . 76, 78, 86, 87, 89–98,	567, 662, 673, 675, 770–772,
106, 173, 174, 319, 325, 336,	774, 776, 778, 781, 782, 817, 821
337, 412, 417, 419–425, 429,	<code>\@glo@shortpluralaccess</code>
466, 467, 537–539, 587, 618,	334, 336, 337, 536–538, 789–791
619, 662, 663, 668, 670–676,	<code>\@glo@sort</code> 14, 16, 23, 24, 74, 93, 96,
680, 718, 719, 791, 792, 840, 872	100, 101, 377, 378, 382, 410,
<code>\@glo@list</code>	422, 424, 426, 427, 626, 628,
98, 425, 676	631, 632, 660, 673, 674, 677, 678
<code>\@glo@long</code> ... 66, 78, 94, 97, 406,	<code>\@glo@sortedinsert</code>
412, 422, 424, 656, 662, 673, 675 216, <u>216</u> , 492, 493, 744, 745
<code>\@glo@longaccess</code>	<code>\@glo@sortentries</code> <u>214</u> , 217–
334, 336, 337, 536–538, 789–791	219, 491, 494, 495, 743, 745, 746
<code>\@glo@longpl</code>	<code>\@glo@sorthandler@case</code>
78, 94, 97, 412, 422, 424, 518, <u>217</u> , 218, 493, 494, 745, 746
520, 522, 524, 525, 528, 529,	<code>\@glo@sorthandler@letter</code>
564, 662, 673, 675, 770, 771, <u>217</u> , 218, 493, 494, 745, 746
773, 774, 776, 778, 781, 782, 817	<code>\@glo@sorthandler@nocase</code>
<code>\@glo@longpluralaccess</code> <u>217</u> , 219, 493, 495, 745, 746
334, 336, 337, 536–538, 789–791	<code>\@glo@sorthandler@word</code>
<code>\@glo@name</code> <u>216</u> , 217, 493, 494, 745
14, 73, 79,	<code>\@glo@sortinghandler</code>
93, 95–97, 377, 410, 413, 422– 214, 216, 491, 493, 743, 745
424, 626, 660, 663, 672, 674, 675	<code>\@glo@sortinglist</code> 214–216,
<code>\@glo@no@assign@sortkey</code>	219, 491–493, 495, 743, 744, 747
.... 192, <u>226</u> , 480, 499, 731, 751	<code>\@glo@sortmacro@case</code> .. <u>218</u> , 494, 746
<code>\@glo@nonumberlist</code> . 77, 411, 412, 662	<code>\@glo@sortmacro@def</code> ... <u>219</u> , 495, 747
<code>\@glo@numfmt</code> .. 206, 487, 571, 739, 824	<code>\@glo@sortmacro@def@do</code> <u>219</u> , 495, 747
<code>\@glo@parent</code>	<code>\@glo@sortmacro@letter</code> <u>218</u> , 494, 746
16, 76,	
93, 95, 96, 100, 101, 216, 378,	
411, 422, 423, 426, 427, 492,	
627, 661, 673, 674, 677, 678, 744	

<code>\@glo@sortmacro@nocase</code>	218, 494, 746	480, 534, 618, 638, 654, 658,
<code>\@glo@sortmacro@standard</code>	679, 680, 719, 731, 732, 787, 872
.....	218, 494, 746	<code>\@glo@useri</code>
<code>\@glo@sortmacro@use</code> ...	219, 495, 747	94, 97, 412, 422, 424, 662, 673, 675
<code>\@glo@sortmacro@word</code> ..	217, 493, 745	<code>\@glo@useriaccess</code>
<code>\@glo@sorttype</code>	334, 336, 337, 536–538, 789–791
.....	195, 220, 221, 226, 481,	<code>\@glo@userii</code>
495, 496, 499, 733, 747, 748, 751		94, 97, 412, 422, 424, 662, 673, 675
<code>\@glo@storeentry</code>	14–16,	<code>\@glo@useriiaccess</code>
100, 376–378, 426, 626, 627, 677		336, 337, 536–538, 789, 790, 792
<code>\@glo@suffix</code>	124,	<code>\@glo@useriii</code>
206, 442, 487, 571, 693, 739, 824		94, 97, 412, 422, 424, 662, 673, 675
<code>\@glo@symbol</code>	66, 74, 93,	<code>\@glo@useriiiaccess</code>
97, 98, 406, 410, 422, 425, 523,		336, 337, 536–538, 789, 790, 792
527, 656, 660, 673, 676, 776, 780		<code>\@glo@useriv</code>
<code>\@glo@symbolaccess</code>	333,	94, 97, 412, 422, 424, 662, 673, 675
336, 536–538, 567, 789–791, 820		<code>\@glo@userivaccess</code>
<code>\@glo@symbolplural</code>	75,
93, 98, 410, 422, 425, 660, 673, 676		335–337, 536–538, 790–792
<code>\@glo@symbolpluralaccess</code>	<code>\@glo@userv</code>
333, 336, 337, 536–538, 789–791		94, 97, 412, 422, 424, 662, 673, 675
<code>\@glo@text</code> ..	74, 93, 96, 97, 134–	<code>\@glo@uservaccess</code>
139, 163–165, 410, 422, 424,		335–337, 536, 537, 539, 790–792
425, 449–452, 462, 463, 524,		<code>\@glo@uservi</code>
566, 587, 660, 673, 674, 676,		94, 97, 412, 422, 424, 662, 673, 675
701–703, 714, 715, 776, 819, 840		<code>\@glo@userviaccess</code>
<code>\@glo@textaccess</code> 333, 336, 535, 537,		335–337, 536, 537, 539, 790–792
538, 564–566, 789–791, 817–820		<code>\@glodesc</code>
<code>\@glo@thislabel</code> ...	99, 426, 676, 677	97, 425, 675
<code>\@glo@thislettergrp</code>	<code>\@glolist@</code>
220–222, 496, 497, 748, 749		95, 423, 674
<code>\@glo@thisvalue</code>	67, 68, 407, 657	<code>\@gloname</code>
<code>\@glo@tmp</code>	69,	97, 424, 675
86, 87, 207, 417, 487, 668, 739		<code>\@glossary@default@style</code> 9, 9, 11,
<code>\@glo@type</code>	9,	211, 233, 261, 374, 375, 490,
16, 75, 93–98, 174, 175, 192,		504, 532, 624, 625, 742, 756, 785
198, 211–216, 219, 220, 224,		<code>\@glossaryentryfield</code>
225, 243, 374, 378, 410, 422–	
425, 467, 468, 479, 483, 484,		99, 100, 101, 426, 427, 677
490–492, 495, 496, 498, 499,		<code>\@glossarysection</code>
508, 519, 521, 522, 524, 526,	
528, 530, 532, 591, 592, 624,		49, 50, 396, 397, 646, 647
627, 660, 672–676, 719, 731,		<code>\@glossarystyle</code>
735, 736, 741–744, 747, 748,		211, 212, 224, 490, 498, 742, 750
750, 751, 760, 771, 773, 775,		<code>\@glossarysubentryfield</code>
777, 779, 781, 783, 785, 844, 845	
<code>\@glo@types</code>	99, 100, 101, 426, 427, 677, 678
33, 61, 62, 68, 70, 105, 175,		<code>\@gls</code>
190, 192, 193, 266, 319, 388,		134, 134, 449, 701
404, 408, 428, 429, 468, 479,		<code>\@gls@</code>
		107, 109, 134,
		134, 327, 329, 430, 431, 449,
		588, 589, 681, 682, 701, 841, 842
		<code>\@gls@automake@immediate</code>
	
		190, 191, 479, 730, 731
		<code>\@gls@link</code> ...
		120, 120, 439, 440, 690
		<code>\@gls@Hcounter</code> ...
		122, 123, 441, 692
		<code>\@gls@ReturnAfterFi</code>
		236, 505, 506, 757
		<code>\@gls@access@display</code> ..
		341, 540, 794

<code>\@gls@actualchar</code>	<code>\@gls@counter</code>
..... 101, 127, 130, 182, 427,	117, 121, 123, 174, 197, 206,
444, 445, 447, 472, 574, 677,	207, 439–442, 467, 483, 487,
678, 695, 696, 698, 723, 724, 827	571, 690–692, 719, 735, 739, 824
<code>\@gls@addpredefinedattributes</code> .	<code>\@gls@counterwithin</code>
..... 55, 178, 12, 13, 376, 625, 626
187, 400, 469, 476, 650, 721, 728	<code>\@gls@ctr</code>
<code>\@gls@adjustmode</code> . 174, 175, 467, 719	53, 398, 649
<code>\@gls@after</code>	<code>\@gls@currentlettergroup</code>
19, 379, 380, 629 220, 222, 495, 497, 747, 749
<code>\@gls@assign@symbol@field</code>	<code>\@gls@debugfalse</code>
23	5, 372, 622
<code>\@gls@assign@symbolplural@field</code> 23	<code>\@gls@debugtrue</code>
<code>\@gls@automake</code>	5, 372, 621, 622
188,	<code>\@gls@declareoption</code> ... 9, 10, 11,
193, 476, 477, 480, 728, 729, 732	18, 21, 22, 28, 31, 32, 34, 35,
<code>\@gls@automake@immediate</code>	38, 39, 374, 375, 378–381, 385,
..... 189, 190, 477–479, 729–731	387, 388, 390, 391, 624, 625,
<code>\@gls@automake@types</code> .. 190, 190, 193	628–631, 635, 637, 638, 640, 641
<code>\@gls@before</code>	<code>\@gls@default</code>
19, 379, 629	111, 432, 683
<code>\@gls@between</code> . 269, 270, 591, 592, 845	<code>\@gls@default@value</code>
<code>\@gls@body</code>	66–68,
165, 463, 714, 715	74, 79, 80, 93, 95–97, 260, 325,
<code>\@gls@checkactual</code>	406, 407, 410, 413, 414, 422–
125, 130,	425, 527, 586, 587, 656, 657,
185, 443, 447, 474, 694, 698, 726	660, 663, 664, 672–675, 780, 840
<code>\@gls@checkboxar</code>	<code>\@gls@defaulttype</code>
125, 129,	69, 69
185, 443, 446, 475, 694, 697, 726	<code>\@gls@deffile</code> 82–84, 415, 416, 665, 666
<code>\@gls@checkedmkidx</code>	<code>\@gls@define@glossaryentrycounter</code>
..... 124–132, 184–186, 442–	12, 12, 40, 225, 227, 376, 392,
448, 473–475, 693–700, 725–727	499, 500, 625, 626, 642, 751, 752
<code>\@gls@checkesc</code> 126, 130, 185, 694, 727	<code>\@gls@define@glossarysubentrycounter</code>
<code>\@gls@checkescactual</code> .. 125, 127, 13, 40, 225, 227, 376,
185, 443–445, 474, 694–696, 726	392, 499, 500, 626, 642, 751, 752
<code>\@gls@checkescbar</code>	<code>\@gls@defsort</code>
125, 128,	14–
185, 443, 445, 475, 694, 696, 726	17, 97, 377, 378, 425, 626–628, 676
<code>\@gls@checkesclevel</code> ... 126, 128,	<code>\@gls@defsortcount</code>
185, 445, 446, 694, 696, 697, 727	14–
<code>\@gls@checkescquote</code>	17, 71, 377, 378, 409, 626–628, 659
..... 125, 127, 185, 186, 443,	<code>\@gls@deprecated@acr@warning</code> ..
444, 474–476, 694, 695, 726, 727	260, 773, 775, 777, 779, 781, 783
<code>\@gls@checklevel</code> . 126, 129, 185,	<code>\@gls@do@acronymsdef</code>
443, 446, 475, 694, 697, 698, 726 18, 18, 40, 72, 378,
<code>\@gls@checkmkidxchars</code>	379, 392, 409, 628, 629, 642, 659
..... 100, 124, 125, 184,	<code>\@gls@do@automake@err</code> ... 34, 35, 36
194, 205, 207, 426, 442, 443,	<code>\@gls@do@glsunset</code> 118, 118, 134–139
474, 480, 486, 488, 571, 677,	<code>\@gls@do@indexdef</code>
693, 694, 726, 732, 738, 739, 824	38–
<code>\@gls@checkquote</code> 125, 126, 184, 443,	41, 72, 391, 392, 409, 641, 642, 659
444, 473, 474, 694, 695, 725, 726	<code>\@gls@do@numbersdef</code>
<code>\@gls@classI</code> .. 178, 179, 469, 470, 721	38,
<code>\@gls@classII</code>	40, 72, 390–392, 409, 641, 642, 659
179, 470, 721	<code>\@gls@do@symbolsdef</code>
<code>\@gls@codepage</code> 38, 72, 390, 409, 640, 659
..... 60, 214, 403, 491, 653, 743	<code>\@gls@do@symbolssdef</code> .. 40, 392, 642
	<code>\@gls@doautomake</code> 35, 35, 36, 191,
	193, 389, 479, 480, 639, 731, 732

<code>\@gls@docheckquotedef</code>	<code>\@gls@get@counterprefix</code> 207, 487, 739
. . . . 184–186, 473–476, 725–728	<code>\@gls@getbody</code> 165, 463, 714
<code>\@gls@docloadedfalse</code> 4, 371, 621	<code>\@gls@getcounter</code> 72, 409, 659
<code>\@gls@docloadedtrue</code> 4, 371, 621	<code>\@gls@getcounterprefix</code>
<code>\@gls@dodeflistparser</code> . 192, 480, 731 206, 207, 485–487, 737–739
<code>\@gls@doentrycounterdef</code>	<code>\@gls@getgrouptitle</code> 194, 232, 232,
. 40, 41, 392, 642	270, 480, 503, 592, 732, 755, 845
<code>\@gls@doentrydef</code> . 438, 439, 689, 690	<code>\@gls@glossary</code> . . . 199, 199, 484, 736
<code>\@gls@dolast</code> 208, 209, 488, 740	<code>\@gls@glslink@hyper@update@hook</code>
<code>\@gls@donext</code> 208, 209, 488, 740 118, 118
<code>\@gls@donext@def</code> . 173, 467, 718, 719	<code>\@gls@gobbleopt</code> 70, 408, 658
<code>\@gls@dosubentrycounterdef</code>	<code>\@gls@grptitle</code> 232, 268,
. 40, 41, 392, 642	270, 503, 591, 592, 755, 844, 845
<code>\@gls@dothiswrite</code>	<code>\@gls@hyp@opt</code>
. . . . 188–190, 477, 478, 729, 730	. . . 87, 88, 109, 110, 119, 120,
<code>\@gls@elem</code> 269, 591, 844	134–162, 241, 242, 327–330,
<code>\@gls@enablesavenonumberlist</code>	418, 419, 431, 439, 449–462,
. 77, 83, 411, 415, 662, 666	506, 507, 588–590, 669, 682,
<code>\@gls@encapchar</code> 128,	690, 701–713, 758, 759, 841–843
129, 182, 206, 208, 445, 446,	<code>\@gls@hyp@opt@cs</code> 119, 439, 690
472, 487, 488, 571, 574, 696,	<code>\@gls@hypergroup</code> . 268, 269, 591, 844
697, 723, 724, 739, 740, 824, 827	<code>\@gls@if@measuring</code> 101, 101
<code>\@gls@entry@count</code>	<code>\@gls@ifinlist</code> . 53, 53, 398, 648, 649
. . . . 108, 109, 430, 431, 681, 682	<code>\@gls@ifnotmeasuring</code> 427, 678
<code>\@gls@entry@field</code> 86, 87, 106, 162,	<code>\@gls@igtype</code> 73, 410, 660
163, 164, 166–172, 337–339,	<code>\@gls@increment@currcount</code>
417, 418, 429, 462–466, 539, 107, 108, 429, 430, 680, 681
540, 668, 680, 714–718, 792, 793	<code>\@gls@indexdef</code> 38, 39, 391, 641
<code>\@gls@esbsdq</code>	<code>\@gls@initnonumberlist</code> 76,
. . . . 124, 125, 183, 442, 443,	77, 77, 94, 411, 422, 661, 662, 673
473, 575, 693, 694, 724, 725, 828	<code>\@gls@islistofacronyms</code> 19, 379, 629
<code>\@gls@expand@fields</code>	<code>\@gls@keylist</code> 368, 564, 817
. 80, 80, 81, 414, 664	<code>\@gls@keymap</code> 77, 84, 84,
<code>\@gls@expandonce</code> 80, 414, 664	85–87, 325, 335, 412, 415–417,
<code>\@gls@extramakeindexopts</code>	537, 586, 662, 666–668, 790, 839
. . . . 186, 191, 476, 479, 728, 731	<code>\@gls@label</code> 194,
<code>\@gls@fetchfield</code>	197, 203, 205, 206, 480, 481,
. 67, 85, 407, 416, 657, 667	483, 486, 487, 732, 735, 737–739
<code>\@gls@field@link</code>	<code>\@gls@labelsfile</code> 33, 388, 638
. 87, 88, 139, 140–154,	<code>\@gls@langmod</code> 31,
352–357, 418, 419, 452–458,	188, 189, 214, 477, 478, 729, 730
552–555, 669, 704–709, 805–808	<code>\@gls@levelchar</code> 101, 128,
<code>\@gls@fieldaccess@display</code>	129, 182, 427, 445, 446, 472,
. 341, 342–344, 541–543, 794–796	574, 678, 696, 697, 723, 724, 827
<code>\@gls@firsttok</code> . . . 220, 221, 496, 748	<code>\@gls@link</code> . . . 120, 121, 134–139,
<code>\@gls@fixbraces</code>	155–162, 350–352, 440, 449–
. . . . 98, 208, 425, 488, 676, 740	452, 458–462, 550, 551, 690,
<code>\@gls@forbidtexext</code>	691, 701–704, 709–714, 803–805
. 69, 71, 408, 658, 659	
<code>\@gls@fulllist</code> 33, 638	

<code>\@gls@link@checkfirsthyper</code>	<code>\@gls@noidxloclist@sep</code>
. 120 , 134 – 195 , 222 , 223 ,
139 , 440 , 449 – 452 , 691 , 701 – 703	481 , 482 , 497 , 498 , 733 , 749 , 750
<code>\@gls@link@label</code>	<code>\@gls@noist@toolate</code> . . . 182 , 183 , 187
121 , 440 , 520 , 525 , 691 , 772 , 778	<code>\@gls@noref@warn</code> 194 , 197 ,
<code>\@gls@link@nocheckfirsthyper</code> . .	220 , 480 , 483 , 496 , 732 , 735 , 748
. 139 , 139 , 155 –	<code>\@gls@notranslatorhook</code> 27
162 , 452 , 458 – 462 , 704 , 709 – 713	<code>\@gls@numberlink</code> . . 235 , 236 , 505 , 757
<code>\@gls@link@opts</code>	<code>\@gls@numbersdef</code> 38 , 38 , 390 , 391 , 641
121 , 440 , 520 , 525 , 691 , 772 , 778	<code>\@gls@numlist@lastsep</code>
<code>\@gls@list</code> 269 , 270 , 591 , 592 , 844 , 845 173 , 467 , 718 , 719
<code>\@gls@listsuffix</code> 53 , 398 , 648	<code>\@gls@numlist@nextsep</code>
<code>\@gls@loadlist</code> 173 , 467 , 718 , 719
. . 11 , 11 , 261 , 375 , 532 , 625 , 785	<code>\@gls@numlist@sep</code> 173 , 467 , 718 , 719
<code>\@gls@loadlong</code> 10 ,	<code>\@gls@old@chapter</code> 41 , 392 , 642
10 , 11 , 261 , 375 , 532 , 624 , 625 , 785	<code>\@gls@oldnewglossaryentryposthook</code>
<code>\@gls@loadsuper</code> 10 , 336 , 537 , 791
10 , 11 , 261 , 375 , 532 , 625 , 785	<code>\@gls@oldnewglossaryentryprehook</code>
<code>\@gls@loadtree</code> 335 , 336 , 537 , 790
. . 11 , 11 , 261 , 375 , 532 , 625 , 785	<code>\@gls@onlypremakeg</code>
<code>\@gls@local@increment@currcount</code> 41 , 41 , 392 , 642 , 643
. 107 , 108 , 429 , 430 , 680 , 681	<code>\@gls@order</code> 188 – 190 , 477 , 478 , 728 – 730
<code>\@gls@loclist</code>	<code>\@gls@org@LT@output</code> . . . 287 , 600 , 854
. 195 , 196 , 221 , 222 , 481 ,	<code>\@gls@org@glsnoidxdisplayloc</code> . .
482 , 496 , 497 , 733 , 734 , 748 , 749 196 , 482 , 734
<code>\@gls@map</code> . . 84 , 85 , 415 , 416 , 666 , 667	<code>\@gls@org@glseeformat</code> 196 , 482 , 734
<code>\@gls@missinglang@warn</code>	<code>\@gls@override@glossary</code>
. . . 21 , 21 , 44 , 381 , 394 , 630 , 644 37 , 37 , 38 , 389 , 390 , 640
<code>\@gls@missingnumberlist</code>	<code>\@gls@patchtabularx</code> . . . 103 , 427 , 678
. 79 , 97 , 413 , 424 , 663 , 675	<code>\@gls@preglossaryhook</code>
<code>\@gls@noaccess</code> 211 , 213 , 489 , 490 , 741 , 742
335 , 341 , 537 , 540 , 541 , 790 , 794	<code>\@gls@prevlevel</code> . . 298 , 299 , 320 –
<code>\@gls@noexpand@field</code> 79	322 , 581 , 582 , 607 , 608 , 619 ,
<code>\@gls@noexpand@fields</code>	620 , 834 , 835 , 860 , 861 , 873 , 874
. 79 , 81 , 413 , 414 , 663 , 664	<code>\@gls@provide@newglossary</code>
<code>\@gls@nohyperlist</code> 69 , 71 , 408 , 658 , 659
. 20 , 20 , 72 , 73 , 121 ,	<code>\@gls@quotechar</code> 126 – 130 ,
380 , 409 , 410 , 440 , 630 , 660 , 691	182 – 184 , 186 , 443 – 447 , 472 –
<code>\@gls@noidx@do</code> 220 , 221 , 496 , 747 , 748	475 , 574 , 695 – 698 , 723 – 727 , 827
<code>\@gls@noidx@getgrouptitle</code>	<code>\@gls@reference</code> 194 , 197 , 223 , 480 ,
. 194 , 232 , 480 , 503 , 732 , 755	481 , 483 , 498 , 732 , 733 , 735 , 750
<code>\@gls@noidx@sanitizesort</code>	<code>\@gls@removedoption</code> . . 25 , 29 , 30 , 36
. 23 , 196 , 382 , 482 , 631 , 734	<code>\@gls@removespaces</code> 236 , 505 , 757
<code>\@gls@noidx@setsanitizesort</code> 26 ,	<code>\@gls@renewglossary</code>
26 , 196 , 384 , 482 , 633 , 634 , 734 187 , 199 , 476 , 484 , 728 , 736
<code>\@gls@noidxloclist@finalsep</code> . . .	<code>\@gls@replacementtext</code>
. 195 , 196 , 482 , 733 , 734 340 , 341 , 540 , 793 , 794
<code>\@gls@noidxloclist@prev</code> 196 , 222 ,	<code>\@gls@requested@languages</code> 29 , 29 , 43
223 , 482 , 497 , 498 , 734 , 749 , 750	<code>\@gls@rest</code> 165 , 463 , 714 , 715
	<code>\@gls@restore@glsllocal</code> . . . 118 , 122

\@gls@restoreat	83, 415, 666	\@gls@tmp	6, 16, 44,
\@gls@roman	56, 400, 401, 572, 651, 825		56, 58, 80, 125, 200, 236, 270,
\@gls@sanitized@tmp	125, 442, 443, 693		373, 378, 393, 401, 414, 442,
\@gls@sanitizedesc			443, 484, 591, 592, 622, 627,
	22, 381, 386, 631, 636		644, 651, 652, 664, 693, 736, 845
\@gls@sanitizename	23, 381, 631	\@gls@tmpb	126,
\@gls@sanitizesort			126, 127–132, 184, 186, 443–
	14, 23, 377, 381, 626, 631		448, 473–475, 694–700, 725–727
\@gls@sanitizesymbol		\@gls@toc	50, 51, 52, 397, 398, 647, 648
	23, 381, 386, 387, 631, 636	\@gls@type	190, 193, 245,
\@gls@save@glsllocal	118, 118, 122		319, 479, 480, 509, 519, 521,
\@gls@saveentrycounter	122,		523, 524, 526, 529, 530, 532,
	122, 174, 441, 467, 691, 692, 719		618, 731, 732, 761, 771, 774,
\@gls@savenonumberlist			775, 777, 779, 782, 783, 785, 872
	76, 76, 77, 196, 411, 661, 662	\@gls@updatechecked	
\@gls@see@noindex			125, 126, 126, 185, 442,
	7, 7, 76, 373, 374, 411, 623, 661		443, 474, 475, 693, 694, 726, 727
\@gls@setacrstyle	29,	\@gls@usetranslator	27, 28,
	40, 386, 387, 391, 392, 636, 642		29, 43, 385, 386, 393, 634, 635, 643
\@gls@setcounter	71, 72, 409, 659	\@gls@value	79, 80,
\@gls@setdefault@glslink@opts			168, 413, 414, 464, 663, 664, 716
	121, 122, 440, 691	\@gls@warnonglossdefined	
\@gls@setsort	14–17, 122, 174, 377,		22, 22, 210, 381, 489, 631, 741
	378, 441, 467, 626–628, 691, 719	\@gls@warnonthe glossdefined	
\@gls@setupshortcuts	40, 391, 392, 642		22, 22, 228, 381, 501, 631, 753
\@gls@setupsort@clear	16	\@gls@write@entrycounts	
\@gls@setupsort@def	15, 377, 627		108, 108, 430, 431, 681
\@gls@setupsort@none	16, 378, 628	\@gls@writedef	82, 83, 84, 415, 665, 666
\@gls@setupsort@standard		\@gls@writeisthook	
	14, 376, 377, 626		181, 183, 183, 472, 473, 723, 725
\@gls@setupsort@use	15, 377, 627	\@gls@x	236, 757
\@gls@sort	222, 497, 749	\@gls@xdy@Lclass@Alpha-page-numbers	
\@gls@sort@A	216, 217, 493, 745		57
\@gls@sort@B	216, 217, 493, 745	\@gls@xdy@Lclass@Appendix-page-numbers	
\@gls@startswithexpandonce			57
	80, 80, 414, 664	\@gls@xdy@Lclass@Roman-page-numbers	
\@gls@storenonumberlist			56
	77, 97, 411, 424, 661, 662, 675	\@gls@xdy@Lclass@alpha-page-numbers	
\@gls@symbolsdef	38, 38, 390, 640		57
\@gls@this	200, 201, 485, 737	\@gls@xdy@Lclass@arabic-page-numbers	
\@gls@thisHloc	207, 487, 739		57
\@gls@thisentryname	33, 638	\@gls@xdy@Lclass@arabic-section-numbers	
\@gls@thisfield	67, 68, 407, 657		57
\@gls@thislabel	65, 208, 209,	\@gls@xdy@Lclass@roman-page-numbers	
	219, 405, 488, 495, 656, 740, 747		56
\@gls@thislang	43	\@gls@xdy@locationlist	56,
\@gls@thislist	173, 467, 718, 719		178, 179, 400, 469, 470, 650, 721
\@gls@thisloc	207, 487, 739	\@gls@xdy@checkbackslash	
\@gls@thisval	85, 416, 667		125, 131, 442, 448, 693, 699, 700
\@gls@title	49, 396, 646		

<code>\@gls@xdycheckquote</code>	<code>\@glslocref</code>
..... 125, 131 , 442, 447, 693, 699	197, 202, 203, 205, 206, 483,
<code>\@gls@xref</code> ... 207, 208, 488, 739, 740	485–487, 571, 735, 737–739, 824
<code>\@glsAlphacompositor</code> 47,	<code>\@glsminrange</code>
47 , 57, 395, 401, 572, 645, 651, 826 177, 177 , 178, 179, 469,
<code>\@glsHlocref</code>	470, 572, 573, 720–722, 825, 826
202, 203, 205, 485, 486, 737, 738	<code>\@glsname</code> 143, 144, 454, 705
<code>\@glsacronymlists</code> 19,	<code>\@glsname@</code> 144, 354 , 454, 552, 705, 806
19 , 20, 61, 243, 245, 266, 379,	<code>\@glsnavhypertarget</code> 268, 268 , 591, 844
380, 404, 508, 509, 519, 521–	<code>\@glsnextpages</code>
524, 526, 528–530, 532, 534, 212, 226 , 490, 500, 742, 751
629, 654, 760, 761, 771, 773–	<code>\@glsnodesc</code> 78 , 93, 95, 97,
775, 777, 779, 781–783, 785, 787	412, 422, 425, 663, 672, 673, 675
<code>\@glsaddkey</code> 86, 87, 417, 668	<code>\@glsnoname</code> 78 , 93, 95, 97,
<code>\@glsaddstoragekey</code>	412, 422–424, 662, 672, 674, 675
..... 85, 86, 417, 667, 668	<code>\@glsnonextpages</code>
<code>\@glsaddxdyattribute</code> .. 54, 399, 649 212, 226 , 490, 499, 742, 751
<code>\@glsdefaultplural</code> 78 , 413, 663	<code>\@glsnumberformat</code>
<code>\@glsdefaultsort</code> 118, 121, 174, 197, 206,
..... 14, 79 , 377, 413, 626, 663	439, 440, 467, 483, 487, 571,
<code>\@glsdesc</code> 144, 145, 454, 706	690, 691, 719, 735, 738, 739, 824
<code>\@glsdesc@</code> 145, 354 , 454, 553, 706, 806	<code>\@glsopenfile</code>
<code>\@glsdescplural</code> .. 145, 146, 454, 706	187, 188 , 198, 476, 484, 728, 735
<code>\@glsdescplural@</code>	<code>\@glsorder</code>
..... 146, 354 , 454, 553, 706, 806	.. 30 , 191, 192, 387, 479, 636, 731
<code>\@glsdisp</code> 138, 138 , 139, 452, 703	<code>\@glspl</code> 136, 450, 702
<code>\@glsentry</code> 33, 105, 108,	<code>\@glspl@</code> 107, 110, 136,
388, 428, 429, 431, 638, 679–682	136 , 328–330, 430, 431, 450,
<code>\@glsentrytitlecase</code>	588–590, 681, 682, 702, 842, 843
..... 168, 168 , 464, 465, 716	<code>\@glsplural</code> 141, 142, 453, 705
<code>\@glsfirst</code> 141, 453, 704	<code>\@glsplural@</code>
<code>\@glsfirst@</code> 141, 353 , 453, 552, 704, 805 142, 353 , 453, 552, 705, 805
<code>\@glsfirstletter</code>	<code>\@glsprefix@record@hook</code> .. 327,
..... 176, 176 , 177, 468, 720	327 , 328–331, 588–590, 841–844
<code>\@glsfirstplural</code> . 142, 143, 453, 705	<code>\@glsreset</code>
<code>\@glsfirstplural@</code>	103, 104 , 107, 427–429, 678–680
143, 353, 353 , 453, 552, 705, 805	<code>\@glssee</code> .. 98, 208, 425, 488, 676, 740
<code>\@glshypernumber</code>	<code>\@glsshowaccsupp</code> 5, 7 ,
..... 235, 235 , 505, 756, 757	333, 372, 373, 535, 622, 623, 788
<code>\@glsisacronymlistfalse</code> 20, 380, 629	<code>\@glsshowtarget</code> 5,
<code>\@glsisacronymlisttrue</code> 20, 380, 629	6, 6 , 372, 373, 448, 621, 622, 700
<code>\@glslink</code> 102, 122, 132 ,	<code>\@glssymbol</code> 146, 147, 455, 706
133, 174, 268, 441, 448, 449,	<code>\@glssymbol@</code>
467, 591, 692, 700, 701, 719, 844 147, 355 , 455, 553, 706, 806
<code>\@glslocalreset</code>	<code>\@glssymbolplural</code> 147, 148, 455, 707
103, 104 , 107, 428, 429, 678–680	<code>\@glssymbolplural@</code>
<code>\@glslocalunset</code> 148, 355 , 455, 553, 707, 806
104, 104 , 107, 428, 429, 679, 680	<code>\@glstarget</code> ... 133, 133 , 229, 269,
	449, 501, 591, 700, 701, 753, 844
	<code>\@glstext</code> 140, 452, 704

<code>\@glstext@</code>	140, 352, 452, 552, 704, 805	<code>\@ifundefined</code>	43, 277, 288, 300, 307, 321,
<code>\@glunset</code>	104, 104, 107, 428, 429, 679, 680		393, 582, 591, 595, 601, 608,
<code>\@gluseri</code>	148, 149, 455, 707		612, 619, 620, 643, 835, 844,
<code>\@gluseri@</code>	149, 355, 455, 554, 707, 807		849, 855, 861, 862, 865, 866, 873
<code>\@gluserii</code>	149, 150, 456, 707	<code>\@ignored@glossaries</code>	33, 72,
<code>\@gluserii@</code>	150, 356, 456, 554, 707, 807		73, 73, 388, 409, 410, 638, 659, 660
<code>\@gluseriii</code>	150, 151, 456, 708	<code>\@input@</code>	213, 491, 742
<code>\@gluseriii@</code>	151, 356, 456, 554, 708, 807	<code>\@istfilename</code>	46, 191, 192, 395, 479, 645, 731
<code>\@gluseriv</code>	151, 152, 456, 708	<code>\@makecol</code>	287, 600, 601, 854
<code>\@gluseriv@</code>	152, 356, 457, 554, 708, 807	<code>\@makeglossaries@warn@noprintglossary</code>	192, 193, 731, 732
<code>\@gluserv</code>	152, 153, 457, 708	<code>\@makeglossary</code>	32, 187, 192, 387,
<code>\@gluserv@</code>	153, 357, 457, 554, 709, 808		388, 476, 479, 637, 638, 728, 731
<code>\@gluservi</code>	153, 154, 457, 709	<code>\@minus</code>	273, 294,
<code>\@gluservi@</code>	154, 357, 457, 555, 709, 808		313, 594, 604, 615, 847, 858, 869
<code>\@glswidestname</code>	319, 319, 320, 321,	<code>\@mkboth</code>	50, 396, 397, 646, 647
	581, 582, 619, 620, 835, 872, 873	<code>\@newglossary</code>	69, 71, 71, 408, 409, 658, 659
<code>\@glswritefiles</code>	36, 198, 389, 483, 639, 735	<code>\@newglossaryentry@defcounters</code>	98,
<code>\@glxtr@doaccsupp</code>	332, 534, 788		99, 106, 425, 426, 429, 676, 680
<code>\@glxtr@record</code>	327, 588, 841	<code>\@newglossaryentryposthook</code>	86,
<code>\@gobble</code>	5, 14–16, 32–34, 84,		87, 98, 99, 325, 336, 417, 425,
	102, 125, 176, 180, 192, 194,		426, 537, 587, 668, 676, 791, 840
	273, 372, 377, 378, 387, 388,	<code>\@newglossaryentryprehook</code>	86, 87, 92, 94, 99, 325, 335,
	415, 416, 427, 442, 468, 470,		336, 417, 421, 422, 426, 537,
	471, 479, 480, 570, 573, 574,		586, 668, 672, 673, 676, 790, 840
	621, 622, 626–628, 637, 638,	<code>\@nil</code>	19, 98, 124–
	666, 667, 678, 693, 719, 720,		126, 165, 184, 185, 206, 208,
	722, 731, 732, 823, 826, 827, 847		220–222, 235, 236, 380, 425,
<code>\@gobblethree</code>	5, 372, 622		442, 443, 463, 474, 475, 487,
<code>\@idxitem</code>	314, 615, 869		488, 496, 497, 505, 571, 629,
<code>\@ifclassloaded</code>	4,		676, 693, 694, 714, 726, 727,
	12, 49, 371, 375, 396, 621, 625, 646		738, 740, 748, 749, 756, 757, 824
<code>\@ifglossaryexists</code>	62,	<code>\@nnil</code>	19, 208, 380, 488, 629, 740
	62, 210, 404, 489, 654, 655, 741	<code>\@no@makeglossaries</code>	32, 192, 195, 197, 387, 388,
<code>\@ifl@t@r</code>	25, 383, 633		480, 481, 483, 637, 731, 733, 735
<code>\@ifnextchar</code>	71, 119, 409, 439, 659, 690	<code>\@no@post@desc</code>	45, 395, 576, 645, 829
<code>\@ifpackageloaded</code>	4, 9, 10, 27–29,	<code>\@nopostdesc</code>	45, 212, 394, 490, 645, 742
	44, 60, 101, 172, 184, 331, 371,	<code>\@one</code>	102
	374, 385, 386, 393, 403, 427,	<code>\@onelevel@sanitize</code>	6,
	466, 473, 534, 621, 624, 634–		23, 33, 56, 84, 100, 124, 125,
	636, 643, 654, 678, 718, 725, 787		176, 177, 181, 207, 209, 221,
<code>\@ifstar</code>	62,		236, 373, 382, 401, 415, 427,
	70, 85, 86, 119, 176, 177, 238,		442, 443, 468, 469, 471, 488,
	404, 408, 417, 439, 468, 506,		489, 496, 572, 573, 622, 631,
	654, 658, 667, 668, 690, 720, 758		

632, 638, 651, 666, 677, 693, 720, 723, 739, 741, 748, 825, 827	\@printgloss@setsort
\@onlypreamble 71, 81, 92, 93, 108, 111, 193, 196, 409, 415, 421, 430, 432, 480, 482, 659, 665, 672, 681, 683, 732, 734 192, 195, <u>211</u> , 212, 480, 481, 489, 490, 731, 733, 741, 742
\@onlypremakeg <u>41</u> , 46, 47, 53, 55, 58, 71, 183, 392, 395, 399, 402, 409, 473, 642, 645, 649, 650, 652, 659, 725	\@printglossary 210, 211, <u>211</u> , 489, 741
\@org@Gls@acrentryname <u>165</u>	\@roman 56, 400, 572, 651, 825
\@org@glossaryentrynumbers 212, 213, 490, 742	\@secondofthree . . . 119, 133, 135, 137, 155, 157, 159, 161, 350, 439, 449–451, 458–460, 462, 550, 690, 701, 703, 710–713, 803
\@org@glsgls@assign@descplural 519, 526, 528, 530, 564, 567, 568, 771, 778, 779, 781, 783, 818, 820, 821	\@secondoftwo 24, 27–29, 44, 84, 85, 102, 133–136, 139, 155, 156, 159, 160, 202, 273, 274, 323, 350–352, 382, 383, 385, 386, 393, 415, 416, 449, 450, 452, 458, 460, 461, 550, 551, 569, 586, 632, 634–636, 643, 666, 667, 700–703, 709, 710, 712, 803–805, 822, 839, 847
\@org@glsgls@assign@firstpl 519–522, 524, 526, 528–530, 564–568, 770, 771, 773–779, 781, 783, 817–821	\@set@glo@numformat <u>124</u> , 206, 442, 487, 571, 693, 739, 824
\@org@glsgls@assign@plural 519–522, 524, 526, 528–530, 564–568, 771, 773–779, 781, 783, 818–821	\@sglsaddkey 86, 417, 668
\@org@glsgls@assign@symbolplural 519–522, 524, 528–530, 565, 566, 568, 771, 773–777, 781, 783, 818, 819, 821	\@sglsaddstoragekey . . . 85, 417, 667
\@org@glsglsnumberformat . 173, 466, 718	\@tabacckludge 24, 382, 632
\@org@newglossaryentryprehook 92, 421, 672	\@text@composite@x 24, 382, 632
\@outputpage 287, 600, 601, 854	\@thirdofthree . . . 119, 136, 138, 156, 158, 160, 162, 351, 439, 450, 451, 458, 460–462, 550, 690, 702, 703, 710–712, 714, 804
\@pglossarysection 49, <u>51</u> , 396, 397, 646, 647	\@this@attr 180, 470, 471, 722
\@pgls 327, <u>327</u> , 588, 841	\@this@childlabel 215, 492, 744
\@pgls@ 327, <u>327</u> , 588, 841	\@this@counter 54, 399, 649
\@pglspl 328, <u>328</u> , 588, 841	\@this@ctr 180, 470, 722
\@pglspl@ 328, <u>328</u> , 588, 841	\@this@key 85, 416, 667
\@plus 273, 294, 313, 594, 604, 615, 847, 858, 869	\@this@label 215, 492, 743, 744
\@print@glossary 210, <u>213</u> , 489, 491, 741, 742	\@this@scs 41, 392, 393, 643
\@print@noidx@glossary 211, <u>219</u> , 489, 495, 741, 747	\@tmp 401, 572, 825
\@printgloss@checkexists . . 210, <u>210</u> , 211, 212, 489, 490, 741, 742	\@use@option 39, 391, 641
\@printgloss@checkexists@allowignored 210, 489, 741	\@warn@nomakeglossaries 190, 214, 479, 491, 730, 743
\@printgloss@checkexists@noignored 210, <u>210</u> , 211, 489, 741	\@wrglossary@pageformat 201, <u>202</u> , 485, 737
	\@wrglossarynumberhook 201, <u>202</u> , 205, 485, 486, 737, 738
	\@write@glslabelsname . . <u>33</u> , 34, 638
	\@xdy@main@language . 31, <u>31</u> , 387, 477, 478, 491, 637, 729, 730, 743
	\@xdy@attributelist <u>52</u> , 54, 180, 398, 399, 470, 648, 649, 722

\@xdyattributes	25, 383, 632
.. 52, 54, 178, 398, 399, 469,	
570, 572, 648, 649, 721, 823, 825	
\@xdycounters	257, 518, 770
..... 53, 54,	
180, 398, 399, 470, 648, 649, 722	
\@xdycrossrefhook	257, 518, 770
..... 57, 179, 401, 470, 652, 722	
\@xdylanguage	333
..... 60, 214, 403, 491, 653, 743	
\@xdylettergroups	\accsuppglossaryentryfield
..... 61, 61, 358, 535, 556, 788, 809
181, 404, 472, 574, 654, 723, 827	\accsuppglossarysubentryfield
\@xdylocationclassorder 358, 535, 556, 788, 809
.. 58, 59,	\Acf
179, 402, 470, 573, 652, 722, 826 257, 518, 770
\@xdylocref	\acf
.. 53, 54, 181, 398, 399, 471, 257, 518, 770
570, 574, 648, 649, 723, 823, 827	\Acfp
\@xdynumbergrouporder 257, 518, 770
..... 61,	\acfp
177, 177, 404, 468, 469, 654, 720 257, 518, 770
\@xdyrequiredstyles	\Acl
..... 59, 59, 257, 518, 770
178, 403, 469, 571, 653, 721, 825	\acl
\@xdysortrules 256, 518, 770
..... 59, 59,	\Aclp
181, 402, 472, 574, 653, 723, 827 257, 518, 770
\@xdystyle	\Aclp
178, 469, 571, 572, 721, 825 256, 518, 770
\@xdyuseralphabets	\Acp
..... 55, 56, 257, 518, 770
178, 400, 469, 572, 650, 721, 825	\acp
\@xdyuserlocationdefs 257, 518, 770
.. 57, 58, 179, 401, 402, 470,	\acrfootnote
570, 573, 651, 652, 722, 824, 826
\@xdyuserlocationnames 258, 519, 520, 525, 772, 778
..... 57,	\ACRfull
58, 401, 402, 571, 651, 652, 824 241, 507, 759
\@xfor@nextelement	\Acrfull
..... 208, 488, 740	.. 241, 257, 507, 518, 758, 770
\@xp@gl@getcounterprefix	\acrfull
.....
..... 203, 205, 206	241, 241, 257, 506, 518, 758, 770
\@xp@glshypernumber	\ACRfullfmt
..... 235, 235
\\	240, 242, 244, 252, 254, 363,
..... 99, 125,	365, 507, 509, 515, 516, 560,
130, 176, 182, 183, 235, 236,	562, 759, 760, 767, 768, 813, 815
426, 442, 468, 472, 473, 505,	\Acrfullfmt
574, 575, 577, 578, 584, 585,
677, 693, 698, 699, 720, 724,	240, 241, 244, 252, 254, 363,
757, 827, 828, 830, 831, 837, 838	365, 507, 508, 515, 516, 560,
\~	562, 759, 760, 767, 768, 813, 815
..... 24, 33, 382, 632, 638	\acrfullfmt
\‘
..... 24, 382, 632	240, 241, 244, 252, 254, 363,
\	365, 506, 508, 515, 516, 560,
..... 125, 128, 130, 185,	561, 758, 760, 766, 768, 813, 815
443, 445, 475, 694, 696, 698, 726	\acrfullformat
\~
..... 24, 383, 632 172, 240, 240, 466, 506,
	507, 518, 531, 718, 758, 770, 784
	\ACRfullpl
 242, 507, 759
	\Acrfullpl
	242, 257, 507, 518, 759, 770
	\acrfullpl

	242, 242, 257, 507, 518, 759, 770
	\ACRfullplfmt

	241, 242, 244, 253, 254, 364,
	366, 508, 509, 515, 516, 560,
	562, 759–761, 767, 768, 813, 815
	\Acrfullplfmt

	241, 242, 244, 252, 254, 364,
A	
\A	232
\a	24, 383, 632
\AA	25, 383, 632

366, 507, 509, 515, 516, 560,
 562, 759, 761, 767, 768, 813, 815
 \acrfullplfmt
 240, 242, 244, 252, 254, 363,
 366, 507, 509, 515, 516, 560,
 562, 759, 761, 767, 768, 813, 815
 \acrlinkfootnote . 258, 258, 519, 772
 \acrlinkfullformat
 240, 240, 241, 506–508, 758–760
 \ACRlong 160, 461, 712
 \Acrlong
 159, 159, 257, 460, 518, 712, 770
 \acrlong . 158, 256, 460, 518, 711, 770
 \ACRlongpl 162, 462, 713
 \Acrlongpl
 161, 161, 257, 461, 518, 713, 770
 \acrlongpl 160, 256, 461, 518, 713, 770
 \acnameformat
 243, 508, 523, 566, 760, 776, 819
 \acnolinkfootnote 258, 519, 772
 acronym (option) 18
 \acronymentry 244,
 244, 246–251, 253–256, 359–
 362, 364–368, 508–518, 556–
 563, 760–770, 810–812, 814–817
 \acronymfont .. 114, 115, 155–158,
 165, 172, 243, 243, 245–256,
 347, 348, 350–352, 359–368,
 436, 437, 458–460, 463, 466,
 508–518, 520, 521, 523–529,
 548–551, 556–563, 565–567,
 687, 688, 709–711, 715, 718,
 760–770, 772, 774, 775, 777–
 780, 782, 801–805, 810–818, 820
 acronymlists (option) 20
 \acronymname 18, 42, 45, 378,
 379, 393, 394, 628, 629, 643, 644
 acronyms (option) 18
 \acronymsort
 244, 245, 246–251, 253–
 256, 359–362, 364, 365, 367,
 368, 508–518, 556–558, 561,
 563, 760–770, 810–812, 814–817
 \acronymtype
 17, 18, 243–245, 378, 379, 508,
 509, 518–532, 564–567, 628,
 629, 760, 761, 770–785, 817–820
 \acrpluralsuffix
 . 239, 244, 246–249, 253–255,
 359, 360, 364–367, 506, 508,
 510–512, 515–522, 524–526,
 528, 529, 531, 556, 557, 561,
 563–568, 758, 760, 762–764,
 767–770, 772–774, 776–779,
 781, 782, 784, 785, 810, 814–821
 \ACRshort 156, 458, 710
 \Acrshort
 155, 155, 256, 458, 518, 710, 770
 \acrshort 154, 256, 458, 518, 709, 770
 \ACRshortpl 158, 459, 711
 \Acrshortpl
 157, 157, 256, 459, 518, 711, 770
 \acrshortpl 156, 256, 459, 518, 710, 770
 acrstyles:
 dua 251, 362
 dua-desc 253, 364
 footnote 253, 365
 footnote-desc 255, 367
 footnote-sc 254, 366
 footnote-sc-desc 255, 367
 footnote-sm 255, 367
 footnote-sm-desc 256, 367
 long-sc-short 248
 long-sc-short-desc 249, 360
 long-short 246, 359
 long-short-desc 249, 360
 long-sm-short 248
 long-sm-short-desc 250, 361
 long-sp-short 247
 long-sp-short-desc 249
 sc-short-long 248
 sc-short-long-desc 250, 361
 short-long 247, 359
 short-long-desc 250, 361
 sm-short-long 249
 sm-short-long-desc 251, 362
 \Acs 256, 518, 770
 \acs 256, 518, 770
 \Acsp 256, 518, 770
 \acsp 256, 518, 770
 \add@accent@ 24, 382, 632
 \addcontentsline 52, 398, 648
 \addglossarytocaptions
 43, 44, 393, 643, 644
 \addtolength .. 321, 582, 620, 835, 873
 \advance 15,
 16, 95, 102, 123, 287, 377, 378,
 423, 441, 600, 627, 674, 692, 854
 \AE 24, 383, 632
 \ae 24, 383, 632

<code>\compatibleglossentry</code>	229 , 332 , 501 , 502 , 535 , 753 , 754 , 788	718 , 719 , 721 , 722 , 726–728 , 732 , 735–737 , 739–744 , 750 , 753–756 , 758 , 771 , 785–788 , 791–793 , 796 , 803–805 , 821– 824 , 835 , 840 , 844 , 845 , 872 , 873
<code>\compatiblesubglossentry</code>	230 , 332 , 502 , 535 , 754 , 788	
<code>\copy</code>	287 , 288 , 600 , 601 , 854	
<code>\count@</code>	221 , 496 , 748	<code>\csshow</code> 266 , 534 , 787
<code>counter</code> (key)	75	<code>\csuse</code> 45 , 48 , 69 , 76 , 79 , 80 , 87 , 88 , 163 , 188–190 , 216 , 218 , 220 , 221 , 223 , 225 , 246 , 325 , 394 , 396 , 408 , 413 , 414 , 418 , 419 , 438 , 439 , 477 , 478 , 492 , 494–496 , 498 , 499 , 504 , 510 , 576–585 , 587 , 644 , 646 , 658 , 663 , 664 , 669 , 689 , 729 , 730 , 744 , 746–748 , 750 , 751 , 756 , 762 , 830–834 , 836–840
<code>counter</code> (option)	20	<code>\csxdef</code> 97 , 108 , 424 , 430 , 675 , 681 , 771
counters:		<code>\currentglossary</code> 12 , 13 , 48 , 212 , 268–270 , 376 , 396 , 490 , 625 , 626 , 646 , 742
<code>glossaryentry</code>	12	<code>\currentglssubentry</code> 13 , 227 , 376 , 500 , 626 , 752
<code>glossarysubentry</code>	13	<code>\CurrentOption</code> 39 , 324 , 331 , 391 , 534 , 586 , 641 , 787 , 839
<code>counterwithin</code> (option)	12	<code>\CurrentTrackedLanguage</code> 44 , 323 , 324 , 394 , 569 , 586 , 644 , 823 , 839
<code>\csdef</code>	22 , 86–88 , 98 , 99 , 106 , 107 , 215 , 216 , 234 , 245 , 246 , 381 , 417–419 , 425 , 426 , 429 , 430 , 492 , 504 , 509 , 510 , 575 , 631 , 668 , 669 , 676 , 677 , 680 , 743 , 744 , 756 , 761 , 829	<code>\CurrentTrackedTag</code> 44 , 323 , 324 , 394 , 569 , 586 , 644 , 822 , 839
<code>\csedef</code>	108 , 201 , 430 , 485 , 681 , 737	<code>\CustomAcronymFields</code> 261 , 531 , 784 , 785
<code>\csgdef</code>	17 , 49 , 69 , 72 , 107 , 108 , 209 , 224 , 396 , 408 , 409 , 429 , 430 , 489 , 498 , 646 , 658 , 659 , 680 , 681 , 741 , 750	<code>\CustomNewAcronymDef</code> 261 , 531 , 532 , 784 , 785
<code>\cslet</code>	76 , 77 , 92 , 99 , 219 , 412 , 421 , 426 , 495 , 662 , 672 , 677 , 747	
<code>\csname</code>	13–16 , 24 , 39 , 42 , 44 , 45 , 50 , 51 , 54 , 56 , 57 , 60 , 62 , 65 , 70–72 , 79 , 80 , 86–88 , 90 , 91 , 95 , 96 , 98–102 , 104 , 105 , 121 , 123 , 124 , 134– 139 , 155–162 , 173 , 174 , 178– 180 , 185–188 , 194 , 197 , 198 , 200 , 201 , 206–208 , 211–213 , 216 , 233 , 234 , 238 , 261–267 , 269 , 319 , 321 , 325 , 336 , 337 , 340 , 344 , 350–352 , 369 , 370 , 376–378 , 382 , 391 , 393 , 394 , 397 , 399 , 401 , 403–405 , 408 , 409 , 413 , 417–420 , 423–428 , 438 , 440–442 , 449–452 , 458– 462 , 467 , 469–471 , 474–476 , 481 , 483–485 , 487 , 488 , 490– 492 , 498 , 501 , 502 , 504 , 506 , 532–535 , 537–540 , 543 , 550 , 551 , 568–571 , 582 , 587 , 591 , 592 , 619 , 620 , 626–628 , 632 , 641 , 643 , 644 , 647 , 649 , 651 , 653 , 654 , 656 , 658 , 659 , 663 , 664 , 667–671 , 674–679 , 689 , 691–693 , 701–703 , 709–714 ,	

D

<code>\d</code>	24 , 382 , 632
<code>\day</code>	178 , 182 , 469 , 472 , 571 , 574 , 721 , 724 , 825 , 827
<code>debug</code> (option)	5
<code>\DeclareAcronymList</code> 18 , 19 , 20 , 243 , 245 , 378–380 , 508 , 509 , 519 , 521 , 522 , 524 , 526 , 528 , 530 , 532 , 628–630 , 760 , 761 , 771 , 773 , 775 , 777 , 779 , 781 , 783 , 785
<code>\DeclareCurrentRelease</code> 2 , 267 , 270 , 273 , 277 , 283 , 288 , 294 , 299 , 307 , 313 , 323 , 324 , 331 , 370 , 371 , 621
<code>\DeclareListParser</code>	192 , 480 , 731

<code>\DeclareOption</code>	9, 324,	<code>\define@boolkey</code>	8, 10–13,
	331, 374, 534, 586, 624, 787, 839		18, 25, 26, 29–31, 36, 118, 226,
<code>\DeclareOptionX</code>	9, 374, 624		374–376, 378, 383, 384, 386,
<code>\DeclareRelease</code>	2, 267, 270, 273,		387, 389, 439, 499, 623–626,
	277, 283, 288, 294, 299, 307,		628, 633, 636, 637, 639, 690, 751
	313, 323, 324, 331, 370, 371, 621	<code>\define@choicekey</code>	5,
<code>\DeclareRobustCommand</code>			7, 8, 13, 26, 28, 30, 34, 37,
 45, 394, 527, 645, 779		39, 76, 224, 225, 372–374, 376,
<code>\def</code>	9, 10, 12–19, 23–25, 30,		384, 385, 387, 388, 390, 411,
	31, 33, 35–38, 40–42, 45, 46,		498, 499, 621, 623, 624, 626,
	49, 53, 55–61, 65, 69–78, 80,		634–636, 639, 640, 661, 750, 751
	90, 92–97, 99, 102, 103, 107–	<code>\define@key</code>	9, 12, 20, 21,
	111, 117, 118, 121–132, 134–		27, 29, 31, 73–78, 86, 87, 117,
	162, 173, 174, 177, 182, 184–		118, 174, 224, 226, 324, 333–
	186, 188–191, 193–196, 198,		335, 374, 376, 380, 384, 387,
	201, 202, 205–208, 211, 212,		410–412, 417, 439, 467, 498,
	214, 219–224, 226, 232–236,		499, 535, 536, 586, 624, 626,
	238, 241–243, 269–272, 298,		630, 634, 637, 660–662, 668,
	299, 319–322, 324, 325, 327–		690, 719, 750, 751, 789, 790, 839
	331, 333–336, 350–357, 374–	<code>\DefineAcronymSynonyms</code>	
	379, 382, 383, 387, 389–396,		. 40, 256, 257, 392, 518, 642, 770
	398, 400–406, 408–412, 414,	<code>\delimN</code>	48, 181, 192, 223,
	420–427, 430–432, 438–462,		236, 396, 471, 480, 497, 505,
	467, 469, 472–479, 481–483,		573, 646, 723, 731, 749, 757, 827
	485–488, 490, 491, 495–499,	<code>\delimR</code>	48, 181, 235, 396,
	503–508, 519–524, 526–528,		471, 505, 573, 646, 723, 757, 827
	530, 532, 535–537, 550–555,	<code>description</code> (key)	73
	564–568, 571, 574, 576, 581,	<code>description</code> (option)	30
	582, 586, 588–593, 607, 608,	<code>descriptionaccess</code> (key)	334
	619, 620, 624–629, 631–633,	<code>\DescriptionDUANewAcronymDef</code> . .	
	637–640, 642–646, 648, 650–	 259, 521, 523, 774, 775
	656, 658–662, 664, 670, 672–	<code>\DescriptionFootnoteNewAcronymDef</code>	
	678, 680–683, 689–714, 718,	 258,
	719, 721, 723, 725–730, 733–		368, 520, 521, 565, 772, 773, 818
	735, 737–743, 747–751, 755–	<code>\descriptionname</code>	
	760, 771, 773–781, 783, 785,		42, 45, 278–282, 284, 285, 290,
	788–791, 803–808, 818–821,		291, 293, 301, 303–306, 308–
	824, 825, 827, 829, 835, 839–		313, 393, 394, 596–599, 602–
	843, 845, 846, 860, 861, 872–874		604, 609–611, 613–615, 643,
<code>\def@glsexdycheckbackslash</code>			644, 849–853, 855–858, 862–869
 131, 132, 447, 448, 699, 700	<code>\DescriptionNewAcronymDef</code>	259,
<code>\DefaultNewAcronymDef</code>	258,		368, 523, 524, 565, 776, 777, 819
	368, 518, 519, 564, 770, 772, 817	<code>descriptionplural</code> (key)	73
<code>\defglstdisplay</code>	117, 438, 689	<code>descriptionpluralaccess</code> (key) . .	334
<code>\defglstdisplayfirst</code>	117, 438, 689	<code>\DH</code>	25, 383, 633
<code>\defglstryfmt</code>	69, 71, 72, 245,	<code>\dh</code>	25, 383, 633
	258, 408, 409, 438, 509, 518,	<code>\dimen@</code>	247, 287, 319,
	519, 521, 523, 525, 527, 529,		511, 600, 618, 619, 763, 854, 872
	531, 658–660, 689, 761, 770,	<code>\directlua</code>	40
	772, 774, 775, 777, 780, 782, 784	<code>\disable@keys</code>	40, 391, 641

<code>disablemakegloss (option)</code>	32	<code>\edef</code>	41, 44, 52–54, 56,
<code>\do</code>	29, 39, 41, 43, 53, 54, 61,		58, 59, 70, 79, 80, 83, 85, 123–
	62, 84, 85, 125, 173, 178–180,		132, 176, 182–186, 188–190,
	190, 192, 193, 200, 208, 215,		192, 198, 214, 216, 217, 221,
	245, 269, 270, 386, 391, 392,		232, 236, 238, 378, 379, 392,
	398, 399, 404, 415, 416, 442,		394, 398–405, 408–410, 413–
	467, 469, 470, 479, 480, 485,		416, 419–422, 426, 438, 440–
	488, 492, 509, 519, 521, 523,		448, 466–468, 472–475, 477,
	524, 526, 529, 530, 532, 571,		478, 480, 483, 486, 487, 489,
	591, 592, 636, 641, 643, 649,		491–493, 496, 500, 503, 505,
	654, 666, 667, 693, 718, 721,		506, 518, 520, 521, 523, 525,
	722, 731, 732, 737, 740, 743,		527, 529, 564–567, 570–572,
	744, 761, 771, 774, 775, 777,		574, 591, 643, 644, 648–653,
	779, 782, 783, 785, 825, 844, 845		658, 659, 663, 664, 666, 667,
<code>\do@glo@storeentry</code>	14–		689, 692–700, 719, 720, 723,
	17, 98, 376–378, 425, 626–628, 676		725–727, 729–731, 735, 738,
<code>\do@glsl@link@checkfirsthyper</code> . .			739, 743, 745, 748, 755, 757,
 120, 122, 134–139,		758, 770, 772, 774, 776, 778,
	155–162, 350–352, 440, 449–		780, 782, 817–820, 823–825, 827
	452, 458–462, 550, 551, 690,	<code>\egroup</code>	24, 92, 173, 206,
	691, 701–704, 709–713, 803, 804		213, 215, 236, 382, 421, 467,
<code>\do@glsl@xdycheckbackslash</code>			490, 492, 632, 672, 719, 742, 744
 124, 125, 442, 693	<code>\else</code>	6, 11–13,
<code>\do@glsl:disablehyperinlist</code>			16–19, 21, 23, 25, 26, 33, 35,
 121, 122, 440, 691		36, 39–41, 46, 47, 49–56, 58–
<code>\do@glslhaschildren</code>			61, 69, 76, 79, 95, 96, 99–102,
 65, 405, 406, 655, 656		107, 108, 118, 121–132, 165,
<code>\do@write@glsl@labelsname</code>			176–178, 181–184, 186–188,
 33, 33, 34, 638		190, 191, 198, 200–202, 205–
<code>\doifglossarynoexistsordo</code>			208, 212, 221, 226–228, 233,
 64, 70, 405, 408, 655, 658		235, 236, 247, 260, 261, 269,
<code>\dtl@ifsingle</code>	503, 755		274, 278, 279, 281, 284, 285,
<code>\dtl@insertinto</code>	216, 493, 745		287, 289, 291, 292, 301, 302,
<code>\dtl@sortresult</code>	216, 217, 493, 745		304, 308, 310, 312, 315, 317,
<code>\dtlcompare</code>	217, 493, 745		318, 320, 321, 341, 372, 373,
<code>\dtlicompare</code>	217, 493, 745		375, 376, 378–381, 383, 384,
<code>\DTLifinlist</code> 73 , 121 , 410 , 440 , 660 , 691			388–392, 395–404, 411, 413,
<code>\DTLifint</code>	232, 503, 755		423, 424, 426, 427, 430, 440–
<code>\dtlletterindexcompare</code> 217 , 493 , 745			452, 463, 468, 469, 471–478,
<code>\DTLsubstituteall</code>	125, 443, 693		483–488, 490, 496, 499–501,
<code>\dtlwordindexcompare</code>	216, 493, 745		503, 505, 511, 521, 522, 524,
<code>dua (acrstyle)</code>	251 , 362		526, 527, 529–532, 540, 571–
<code>dua (option)</code>	30		576, 580–582, 591, 594, 596,
<code>dua-desc (acrstyle)</code>	253 , 364		597, 599–603, 608–610, 612–
<code>\DUANewAcronymDef</code>			614, 616–620, 622, 625–631,
 260, 529, 530, 782, 783		633, 634, 638–643, 645–654,
			661, 663, 674, 675, 677, 678,
			681, 691–703, 714, 715, 720,
			721, 723, 725–730, 735–740,
			742, 748, 751–753, 755, 757,
E			
<code>\eappto</code>	409, 410, 426, 485		

763, 773–775, 777, 779–782,	205, 236, 371, 373, 484, 486,
784, 785, 794, 824, 825, 827–	505, 621, 623, 638, 736, 738, 757
829, 833–835, 844, 847, 849–	<code>\endhead</code> 278–282,
857, 862–864, 866–868, 870–874	284, 285, 289–291, 293, 596–
<code>\emph</code> 208, 237, 488, 506, 740, 758	599, 602–604, 849–853, 855–858
<code>\empty</code> . . . 236, 332, 505, 534, 757, 788	<code>\endtheglossary</code> 4, 371, 621
<code>\encodingdefault</code> 24, 382, 632	<code>entrycounter</code> (option) <u>12</u>
<code>\end</code> 180, 220, 274,	<code>\entryname</code>
277–280, 282, 283, 285–313,	<u>42</u> , 45, 278–282, 284, 285, 290,
471, 496, 573, 594–598, 600–	291, 293, 301, 303–306, 308–
615, 722, 747, 826, 847, 849–869	313, 393, 394, 596–599, 602–
<code>\end@doifinlist</code> 53, 398, 648	604, 609–611, 613–615, 643,
<code>\end@getprefix</code> 207, 487, 739	644, 849–853, 855–858, 862–869
<code>\end@glsl@islistofacronyms</code>	environments:
. 19, 379, 380, 629	<code>theglossary</code> <u>228</u>
<code>\EndAccSupp</code> 332, 535, 788	<code>\epreto</code> 102
<code>\endcsname</code> 13–16, 24,	<code>\equal</code> 27, 41, 51, 123, 192, 269, 384,
39, 42, 44, 45, 50, 51, 54, 56,	392, 397, 441, 479, 503, 591,
57, 60, 62, 65, 70–72, 79, 80,	634, 642, 647, 692, 731, 755, 844
86–88, 90, 91, 95, 96, 98–102,	<code>esclocations</code> (option) <u>10</u>
104, 105, 121, 123, 124, 134–	<code>\expandafter</code> 6, 14–16, 23, 24,
139, 155–162, 173, 174, 178–	39, 41, 44, 54, 56, 57, 59–62,
180, 185–188, 194, 197, 198,	65, 70–73, 79, 80, 84–88, 90,
200, 201, 206–208, 211–213,	91, 95, 96, 98, 100–105, 116,
216, 233, 234, 238, 261–267,	117, 121, 123–131, 163–165,
269, 319, 321, 325, 336, 337,	168, 173, 176, 180, 184–187,
340, 344, 350–352, 369, 370,	198, 200–202, 205, 206, 209,
376–378, 382, 391, 393, 394,	212, 216, 221, 222, 229, 234,
397, 399, 401, 403–405, 408,	236, 238, 261–267, 269, 270,
409, 413, 417–420, 423–428,	319, 325, 336, 337, 340, 341,
438, 440–442, 449–452, 458–	368–370, 373, 377, 378, 382,
462, 467, 469–471, 474–476,	391–394, 399, 401–405, 408–
481, 483–485, 487, 488, 490–	410, 413–420, 423–428, 438,
492, 498, 501, 502, 504, 506,	440–448, 463, 467, 468, 470,
532–535, 537–540, 543, 550,	471, 473–476, 483–488, 490,
551, 568–571, 582, 587, 591,	492, 496, 497, 501, 502, 504–
592, 619, 620, 626–628, 632,	506, 520, 525, 532–535, 537–
641, 643, 644, 647, 649, 651,	541, 564, 568–571, 573, 574,
653, 654, 656, 658, 659, 663,	587, 591, 592, 619, 622, 626–
664, 667–671, 674–679, 689,	628, 631, 632, 641, 643, 644,
691–693, 701–703, 709–714,	649, 651, 653, 654, 656, 658–
718, 719, 721, 722, 726–728,	660, 663, 664, 666–671, 674–
732, 735–737, 739–744, 750,	679, 689, 691–699, 714–716,
753–756, 758, 771, 785–788,	719, 720, 722, 725–728, 735–
791–793, 796, 803–805, 821–	740, 742, 744, 748, 749, 753,
824, 835, 840, 844, 845, 872, 873	754, 756–758, 771, 772, 778,
<code>\endfoot</code> 278–280, 282,	785–788, 791–794, 817, 821–
284, 285, 289–291, 293, 596–	824, 826, 827, 840, 844, 845, 872
599, 602–604, 849–853, 855–858	<code>\expandonce</code> 79, 80, 91, 125, 126, 185,
<code>\endgroup</code> 4, 6, 33, 200,	186, 201, 216, 217, 244, 413,

414, 442, 443, 474–476, 485, 493, 501, 502, 508, 518, 520, 522, 524, 525, 528, 529, 535, 663, 664, 671, 693, 694, 726, 727, 737, 745, 753, 754, 760, 770–774, 776, 778, 781, 782, 788		776, 778, 780, 803, 809, 810, 814
<code>\ExplSyntaxOff</code>	3, 232, 233	<code>firstplural</code> (key) 74
<code>\ExplSyntaxOn</code>	2, 232, 233	<code>firstpluralaccess</code> (key) 333
F		<code>\fmtversion</code> 25, 383, 633
<code>\fi</code> . . . 4–6, 8, 9, 11–14, 16–19, 21, 23, 25, 26, 28–30, 32, 33, 35, 36, 38–41, 45–47, 50–61, 69, 72, 76, 79, 91, 95–97, 99–102, 107, 108, 118, 121–132, 165, 175–178, 181, 183, 184, 186, 187, 189–193, 198–203, 205– 209, 212, 214, 221, 225–228, 233, 235, 236, 247, 257, 260, 261, 269, 274, 278, 279, 281, 284, 285, 287–289, 291, 292, 301, 302, 304, 308, 310, 312, 315–321, 332, 341, 371–384, 386, 388–392, 394–404, 409, 411, 413, 423–427, 430, 440– 452, 463, 467–469, 471, 473– 477, 479, 480, 484–491, 496, 499–501, 503–506, 511, 518, 519, 521, 522, 524–532, 534, 535, 541, 570–576, 580–582, 591, 594, 596, 597, 599–603, 608–610, 612–614, 616–631, 633–635, 638–643, 645–654, 659, 661, 663, 671, 673–678, 681, 691–704, 715, 719–721, 723, 725–732, 735–743, 748, 751–753, 755–757, 763, 770, 771, 773–775, 777, 779–785, 787, 788, 794, 823–825, 827– 829, 833–835, 844, 847, 849– 857, 862–864, 866–868, 870–874		<code>\footnote</code> 253, 254, 258, 365, 516, 519, 561, 768, 772, 814
<code>\findrootlanguage</code>	60, 403, 653	<code>footnote</code> (acrstyle) 253 , 365
<code>first</code> (key)	74	<code>footnote</code> (option) 29
<code>firstaccess</code> (key)	333	<code>footnote-desc</code> (acrstyle) . . . 255 , 367
<code>\firstacronymfont</code>	116, 243 , 246–248, 253, 254, 349, 350, 359, 360, 365, 438, 508, 510, 511, 516, 520, 523, 525, 527, 550, 556, 557, 561, 688, 689, 760, 762, 763, 768, 772,	<code>footnote-sc</code> (acrstyle) 254 , 366
		<code>footnote-sc-desc</code> (acrstyle) . 255 , 367
		<code>footnote-sm</code> (acrstyle) 255 , 367
		<code>footnote-sm-desc</code> (acrstyle) . 256 , 367
		<code>\FootnoteNewAcronymDef</code> . . . 259 , 368 , 525, 526, 566, 778, 779, 820
		<code>\footnotesize</code> 6, 373, 622
		<code>\forallacronyms</code> 61 , 404, 654
		<code>\forallglossaries</code>
		. . . 61 , 62, 198, 211, 319, 404, 483, 489, 618, 654, 735, 741, 872
		<code>\forallglsentries</code> . . . 33, 62 , 105, 108, 175, 388, 404, 428, 429, 431, 468, 638, 654, 679–681, 719
		<code>\ForEachTrackedDialect</code> . 44, 323, 324, 394, 569, 586, 644, 822, 839
		<code>\forglsentries</code> 61 , 62, 65, 99, 219, 319, 404, 405, 426, 495, 618, 654, 656, 677, 747, 872
		<code>\forlistcsloop</code>
	 215, 220, 492, 496, 743, 747
		<code>\forlistloop</code>
	 196, 222, 482, 497, 734, 749
		G
		<code>\gdef</code> 15, 33, 54, 70, 90, 95, 96, 199, 226, 269, 377, 399, 408, 419, 423, 484, 499, 500, 591, 627, 638, 649, 658, 670, 674, 736, 751, 844
		<code>\Genacrfullformat</code>
	 116, 116 , 244, 246–248, 254, 349, 350 , 359, 360, 365, 437, 438, 509–511, 516, 549, 550, 556, 557, 561, 688, 760– 763, 768, 802, 803, 809, 810, 814
		<code>\genacrfullformat</code>
	 115, 116, 116 , 244, 246– 248, 253, 349, 349 , 359, 365, 437, 438, 508–511, 516, 549,

556, 557, 561, 688, 689, 760–
763, 768, 802, 803, 809, 810, 814

`\GenericAcronymFields`
..... 244, [244](#), 246–253,
255, 256, 359–365, 367, 368,
508–518, 556–558, 560, 561,
563, 760–769, 809–814, 816, 817

`\Genplacrformat`
. 115, [117](#), 244, 246–248, 254,
349, [350](#), 359, 360, 365, 437,
438, 509–511, 516, 549, 550,
556, 557, 561, 688, 689, 761–
763, 768, 802, 803, 809, 810, 814

`\genplacrformat` 115,
[116](#), [117](#), 244, 246–248, 254,
349, [350](#), 359, 360, 365, 437,
438, 509–511, 516, 549, 550,
556, 557, 561, 688, 689, 761–
763, 768, 802, 803, 809, 810, 814

`\GetTitleStringDisableCommands`
..... 273, 847

`\GetTitleStringSetup` 273, 847

`\glo@desc` 576, 829

`\glo@do@compare` .. 216, 217, 493, 745

`\glo@grabfirst`
.... [220](#), 222, 496, 497, 748, 749

`\glo@label` . 65, 99, 405, 426, 656, 677

`\glo@list` 99, 426, 677

`\glo@name` 229, 501, 753

`\glo@parent` 65, 405, 656

`\glo@type` 99, 426, 677

`\glo@value` 84, 415, 666

`\global` 15, 16, 79,
82, 92, 98, 104, 105, 200, 213,
221, 222, 226, 287, 288, 377,
378, 413, 415, 421, 425, 428,
484, 490, 496, 497, 500, 601,
627, 628, 663, 665, 672, 676,
679, 736, 742, 748, 749, 752, 854

`\glo@linkprefix` ... 122, [122](#), 174,
229, 441, 467, 501, 692, 719, 753

`\glosortentrieswarning` 21,
[21](#), 214, 380, 381, 491, 630, 743

`\GlossariesWarning`
.. 5, 7, 21, [21](#), 22, 25–27, 32–
34, 37, 43, 49, 52, 64, 68, 69,
76, 79, 82, 83, 109–111, 172,
187–190, 194–197, 200, 202,
207, 210, 211, 260, 332, 372,
374, 380, 381, 383, 384, 387–
390, 396, 398, 405, 407, 411,
413, 431, 432, 438, 439, 466,
477, 478, 480–484, 487, 489,
490, 502, 504, 534, 570, 621–
623, 630, 631, 633, 634, 637–
640, 646, 648, 655, 658, 661,
663, 665, 666, 682, 689, 690,
718, 729, 730, 732–736, 739,
741, 742, 754, 756, 783, 788, 823

`\GlossariesWarningNoLine`
..... 5, 21, [21](#), 193, 194,
198, 214, 269, 372, 380, 381,
479–481, 483, 491, 591, 621,
622, 630, 732, 733, 735, 743, 844

`\glossary` 36,
37, [37](#), 389, 390, 571, 639, 640, 824

`\glossaryentry` 206,
208, 487, 488, 571, 739, 740, 824

`glossaryentry (counter)` [12](#)

`\glossaryentryfield` [230](#), 358, 501,
502, 556, 576–581, 583–585,
753, 754, 809, 829–834, 836–838

`\glossaryentrynumbers`
..... [9](#), 10, 180, 212,
213, 222, 226, 374, 375, 471,
490, 497, 499, 500, 573, 624,
723, 742, 748, 749, 751, 752, 826

`\glossaryheader`
. 180, 220, [229](#), 271, 274–276,
278–282, 284, 285, 289–293,
295–298, 300, 302, 303, 307,
309, 311, 314, 316–318, 320,
322, 471, 496, 501, 573, 592,
594–599, 601–610, 612–614,
616–620, 722, 747, 753, 826,
845, 847–853, 855–864, 866–874

`\glossarymark` .. 49, [50](#), 396, 646, 647

`\glossaryname` 17, [42](#),
44, 45, 378, 393, 394, 628, 643, 644

`\glossarypostamble`
..... [49](#), 180, 220, 396,
471, 496, 573, 646, 722, 748, 826

`\glossarypreamble` [48](#), 180, 220, 396,
470, 495, 573, 646, 722, 747, 826

`\glossarysection` [49](#), 180, 220, 396,
470, 495, 573, 646, 722, 747, 826

`\glossarystyle` [233](#), 504, 756

`glossarysubentry (counter)` [13](#)

<code>\glossarysubentryfield</code>	<code>\GLS</code>
. 231 , 358, 502, 503,	<code>\Gls</code> 109 , 134 , 134 , 238, 257, 431,
556, 576–585, 754, 809, 829–838	449, 506, 518, 682, 701, 758, 770
<code>\glossarytitle</code>	<code>\gls</code>
180, 211, 212,	37, 109, 134,
220, 224, 470, 490, 495, 498,	134 , 194, 228, 238, 257, 390,
573, 722, 741, 742, 747, 750, 826	431, 449, 480, 500, 506, 518,
<code>\glossarytoctitle</code>	640, 682, 701, 732, 752, 758, 770
9,	<code>\gls@delimN</code>
17, 18, 38, 42, 45, 49, 180,	236
211, 220, 224, 225, 374, 378,	<code>\gls@accessibility</code>
379, 390, 391, 393, 394, 396, 332 , 333, 535, 788, 789
470, 490, 495, 498, 499, 573,	<code>\gls@accsupp@engine</code> 332, 332 , 535, 788
624, 628, 629, 641, 643, 644,	<code>\gls@Alphpage</code>
646, 722, 741, 747, 750, 751, 826	200, 201 , 205, 485, 486, 736–738
<code>\glossentry</code>	<code>\gls@alphpage</code>
99,	200, 201 , 205, 485, 486, 736–738
213, 222, 229 , 271, 274–279,	<code>\gls@arabicpage</code>
281, 289, 290, 292, 300, 302,	200, 201 , 205, 485, 486, 736–738
304, 307, 309, 311, 315, 316,	<code>\gls@assign@desc</code>
318, 320, 426, 490, 497, 502, 92 , 92 , 97, 421, 425, 672, 676
592, 594–597, 601–603, 608–	<code>\gls@assign@desc@field</code>
610, 612–614, 616–619, 677,	22
742, 749, 754, 845, 847–851,	<code>\gls@assign@descplural</code> 519, 526,
855–857, 862–864, 866–871, 873	528, 530, 564, 567, 568, 771,
<code>\Glossentrydesc</code>	778, 779, 781, 783, 818, 820, 821
. 230 , 358, 502, 555, 754, 808	<code>\gls@assign@descplural@field</code>
<code>\glossentrydesc</code>	23
. 230 , 230 , 271, 272, 274–	<code>\gls@assign@field</code>
276, 278, 279, 281, 289, 290,	81, 81 , 86, 87, 92, 94, 96–98,
292, 300, 302, 304, 307–309,	325, 414, 417, 421, 422, 424,
311, 315, 316, 318, 320, 321,	425, 587, 664, 668, 672–676, 840
358, 502, 555, 593–597, 601–	<code>\gls@assign@firstpl</code> 519–522, 524,
603, 608–610, 612–614, 616–	526, 528–530, 564–568, 770,
620, 753, 808, 846–851, 855–	771, 773–779, 781, 783, 817–821
857, 862–864, 866–868, 870–873	<code>\gls@assign@name@field</code>
<code>\Glossentryname</code>	23
229, 501, 753	<code>\gls@assign@plural</code>
<code>\glossentryname</code> 229, 229 , 231, 271–	519–522,
279, 281, 289, 290, 292, 300,	524, 526, 528–530, 564–568,
302, 304, 307, 309, 311, 315,	771, 773–779, 781, 783, 818–821
316, 318, 320, 321, 357, 358,	<code>\gls@assign@symbolplural</code>
501, 555, 592–597, 601–603, 519–522, 524,
608–610, 612–614, 616–620,	528–530, 565, 566, 568, 771,
753, 808, 846–851, 855–857,	773–777, 781, 783, 818, 819, 821
862–864, 866–868, 870, 871, 873	<code>\gls@assign@type@field</code>
<code>\Glossentrysymbol</code>	22
. 230 , 358, 502, 555, 754, 809	<code>\gls@automake@makegloss</code>
<code>\glossentrysymbol</code> 230, 230 , 271, 34 , 34 , 35, 36, 191
272, 281, 292, 304, 311, 315,	<code>\gls@automake@nr</code>
316, 318, 320, 321, 358, 502,	34 , 35, 36, 190, 388, 479, 639, 730
555, 592, 593, 597, 603, 610,	<code>\gls@automake@option</code>
614, 616–620, 754, 809, 846,	34 , 34 , 35
851, 857, 864, 868, 870, 871, 873	<code>\gls@automake@val</code>
	34 , 35, 388, 639
	<code>\gls@begindocdefs</code>
	83, 83 , 415, 666
	<code>\gls@checkisacronymlist</code>
 20 , 120, 380, 440, 629, 691

<code>\gls@checkseeallowed</code>	<code>\gls@inlinesubsep</code>
..... 75, <u>75</u> , 192, 194,	271, 272, 576, 592, 593, 829, 845, 846
411, 414, 479, 480, 661, 731, 732	<code>\gls@islistofacronyms</code>
<code>\gls@checkseeallowed@preambleonly</code> 19, 379, 380, 629
..... <u>76</u> , 411, 414, 661	<code>\gls@istfilebase</code> 45, 46, <u>46</u> , 188,
<code>\gls@codepage</code> 31, 60, 188,	189, 395, 477, 478, 645, 729, 730
189, 214, 387, 403, 477, 478,	<code>\gls@kernelglossredefs@nr</code> 37
491, 637, 653, 654, 729, 730, 743	<code>\gls@kernelglossredefs@val</code> 37
<code>\gls@debug@nr</code> .. 5, 372, 390, 621, 640	<code>\gls@label</code> 238, 506, 758
<code>\gls@debug@val</code> . 5, 372, 390, 621, 640	<code>\gls@level</code> <u>79</u> , 95, 96, 221,
<code>\gls@defdocnewglossaryentry</code> ...	413, 423, 496, 497, 663, 674, 748
..... <u>81</u> , 83,	<code>\gls@local@stepcounter</code> ... 102, <u>102</u>
106, 414, 415, 429, 665, 666, 680	<code>\gls@measuring@reset</code> 102
<code>\gls@defglossaryentry</code> ... 81–83,	<code>\gls@mfirstuc@nr</code> 39
92, <u>93</u> , 414, 415, 421, 665, 666, 672	<code>\gls@mfirstuc@val</code> 39
<code>\gls@delimN</code> 235, <u>236</u>	<code>\gls@newacronym</code> 238, <u>238</u>
<code>\gls@delimR</code> 235, <u>235</u>	<code>\gls@noidx@nonumberlist</code> ... <u>76</u> , 196
<code>\gls@disablepagerefexpansion</code> ..	<code>\gls@noidxglossary</code>
200, <u>200</u> , 205, 484–486, 736–738 194, <u>197</u> , 480, 483, 732, 735
<code>\gls@do@addxdyattribute</code> 54, 399, 649	<code>\gls@nonumberlist@nr</code> .. 76, 411, 661
<code>\gls@docdef@addlabel</code>	<code>\gls@nonumberlist@val</code> . 76, 411, 661
..... 82, <u>82</u> , 83, 665, 666	<code>\gls@nosetquote</code>
<code>\gls@docdef@labellist</code> .. 82, <u>82</u> , 665 93, 182, 183, 186, 421,
<code>\gls@docclearpage</code>	472, 473, 476, 672, 723, 725, 728
..... <u>51</u> , 52, 397, 398, 647, 648	<code>\gls@number</code> 205, 486, 738
<code>\gls@doentryfmt</code> <u>69</u> , 408, 658	<code>\gls@numberedsection@nr</code>
<code>\gls@dosubst</code> 125, 443, 693, 694	.. 8, 224, 374, 498, 499, 624, 750
<code>\gls@dotoc@title</code>	<code>\gls@numberedsection@val</code>
.... 212, 224, 490, 498, 742, 750 8, 224, 374, 498, 624, 750
<code>\gls@end@sanitizesort</code>	<code>\gls@numberpage</code>
..... 23, 24, 382, 631, 632	200, <u>201</u> , 205, 485, 486, 736–738
<code>\gls@endcheck</code> 80, 414, 664	<code>\gls@org@glossaryentryfield</code> ...
<code>\gls@glossary</code> 213, 490, 742
.. 37, 38, <u>198</u> , 206–208, 390,	<code>\gls@org@glossarysubentryfield</code>
484, 487, 488, 640, 736, 739, 740 213, 490, 742
<code>\gls@glsdefs@addlabel</code> <u>82</u> , 83, 665, 666	<code>\gls@org@insert</code>
<code>\gls@glsdefs@labellist</code> . 82, <u>82</u> , 665	523, 525, <u>527</u> , 775, 777, 778, 780
<code>\gls@gobbleopt</code>	<code>\gls@org@label</code> 101, 102
..... <u>70</u> , 71, 408, 409, 658, 659	<code>\gls@org@link</code> 102
<code>\gls@grplabel</code> 268, 591, 844	<code>\gls@org@refstepcounter</code> 101, <u>101</u> , 102
<code>\gls@hypergroup@prerun</code>	<code>\gls@org@target</code> 102
..... 269, <u>269</u> , 591, 844	<code>\gls@org@Alpha</code> 204, 205, 486, 738
<code>\gls@if@measuring</code> <u>101</u> , 102	<code>\gls@org@alpha</code> 204, 205, 486, 738
<code>\gls@ifnotmeasuring</code> <u>102</u> ,	<code>\gls@org@arabic</code> ... 204, 205, 486, 738
103, 104, 200, 427, 428, 678, 679	<code>\gls@org@number</code> ... 204, 205, 486, 738
<code>\gls@inlinepostchild</code> 271,	<code>\gls@org@Roman</code> 204, 205, 486, 738
272, 576, 592, 593, 829, 845, 846	<code>\gls@org@romannumeral</code>
<code>\gls@inlinesep</code> 204, 205, 486, 738
271, 576, 592, 593, 829, 845, 846	<code>\gls@org@the</code> 204, 205, 486, 738

<code>\gls@original@glossary</code>	<code>\glsaccessdisplay</code> 344 , 543 , 796
. 36 , 37 , 389 , 390 , 639 , 640	<code>\glsaccessibility</code>
<code>\gls@original@makeglossary</code> 332 , 341 , 535 , 540 , 788 , 793
. 36 , 38 , 389 , 390 , 639 , 640	<code>\glsaccsupp</code> 340 , 341 , 341 , 540 , 793 , 794
<code>\gls@protected@pagefmts</code> 125 , 200 ,	<code>\glsacronymtrue</code> 18 , 379 , 629
200 , 201 , 442 , 485 , 693 , 736 , 737	<code>\glsacrpluralsuffix</code>
<code>\gls@Romanpage</code> 43 , 43 , 239 , 248 , 249 , 253 –
200 , 201 , 205 , 485 , 486 , 736 – 738	255 , 393 , 506 , 512 , 515 – 517 ,
<code>\gls@romanpage</code>	519 , 643 , 758 , 764 , 767 – 769 , 772
200 , 201 , 205 , 485 , 486 , 736 – 738	<code>\glsacrshortcutsfalse</code> 40 , 392 , 642
<code>\gls@save@numberlist</code> 9 ,	<code>\glsacrshortcutstrue</code> 40 , 391 , 642
10 , 209 , 374 , 375 , 489 , 624 , 740	<code>\glsacspace</code> 247 ,
<code>\gls@seenoinindex@nr</code> 7 , 373 , 623	247 , 249 , 510 , 511 , 513 , 762 – 764
<code>\gls@seenoinindex@val</code> 7 , 373 , 623	<code>\glsadd</code> 37 , 174 ,
<code>\gls@set@xr@key</code> 75 , 75 , 411 , 661	174 , 175 , 390 , 467 , 468 , 640 , 719
<code>\gls@start@measuring</code> 102 , 102 , 103	<code>\glsaddall</code> 175 , 468 , 719
<code>\gls@stop@measuring</code> 102 , 102 , 103	<code>\glsaddallunused</code> 175 , 468 , 719
<code>\gls@suffixF</code> 47 ,	<code>\glsaddkey</code> 86 , 417 , 668
47 , 181 , 183 , 395 , 471 , 473 ,	<code>\GlsAddLetterGroup</code> 61 , 404 , 654
573 , 575 , 645 , 723 – 725 , 827 , 828	<code>\glsaddprotectedpagefmt</code>
<code>\gls@suffixFF</code> 47 , 201 , 202 , 485 , 737
47 , 181 , 183 , 395 , 471 , 473 ,	<code>\GlsAddSortRule</code> 59 , 402 , 403 , 653
573 – 575 , 645 , 723 , 725 , 827 , 828	<code>\glsaddstoragekey</code> 85 , 417 , 667
<code>\gls@text</code> 116 , 117 , 438 , 689	<code>\GlsAddXdyAlphabet</code> 55 , 400 , 650
<code>\gls@the</code> 205 , 486 , 738	<code>\GlsAddXdyAttribute</code> 53 , 54 , 54 , 55 ,
<code>\gls@thissty</code> 29 , 386 , 636	399 , 400 , 570 , 649 , 650 , 823 , 824
<code>\gls@tmp</code> 190 ,	<code>\GlsAddXdyCounters</code>
193 , 198 , 260 , 483 , 527 , 735 , 780 53 , 53 , 54 , 55 , 72 , 398 ,
<code>\gls@tmplen</code> 132 , 319 , 321 , 448 ,	399 , 409 , 570 , 648 – 650 , 659 , 823
582 , 618 – 620 , 700 , 835 , 872 , 873	<code>\GlsAddXdyLocation</code>
<code>\gls@tr@set@acronym@toctitle</code> 57 , 401 , 402 , 570 , 652 , 824
. 18 , 379 , 628 , 629	<code>\GlsAddXdyStyle</code> 59 , 403 , 653
<code>\gls@tr@set@main@toctitle</code>	<code>\glsautomakefalse</code> 34 ,
. 17 , 378 , 628	35 , 191 , 388 , 389 , 479 , 639 , 731
<code>\gls@tr@set@numbers@toctitle</code> . .	<code>\glsautomaketrue</code> 34 , 35 , 388 , 639
. 38 , 391 , 641	<code>\glsautoprefix</code>
<code>\gls@tr@set@symbols@toctitle</code> 8 , 9 , 225 , 374 , 499 , 624 , 751
. 38 , 390 , 641	<code>\glsbackslash</code> 176 , 468 , 720
<code>\gls@translate@nr</code> 28 , 385 , 635	<code>\glscapitalisewords</code> 168 , 168 , 716
<code>\gls@translate@val</code> 28 , 385 , 635	<code>\glscapscase</code> 112 –
<code>\gls@wrglossary</code> 199 , 200 , 484 , 736	115 , 134 – 139 , 155 – 162 , 251 ,
<code>\gls@xdy@set@lang</code> 31 , 188 , 189 , 214	252 , 345 – 352 , 362 , 363 , 432 ,
<code>\gls@xdystring</code>	433 , 435 – 437 , 449 – 452 , 458 –
. 124 , 125 , 442 , 443 , 693 , 694	462 , 514 , 523 , 527 , 543 , 544 ,
<code>\gls@xindy@glslnumbersfalse</code>	546 – 551 , 559 , 683 , 684 , 686 –
. 32 , 387 , 637	688 , 701 – 703 , 709 – 714 , 766 ,
<code>\gls@xindy@glslnumberstrue</code>	776 , 780 , 796 , 797 , 799 – 805 , 812
. 31 , 387 , 637	<code>\glscategory</code> 340 , 540 , 793
<code>\gls@xr@key</code> 7 ,	<code>\glsclearpage</code> 51 , 52 , 397 , 398 , 647 , 648
75 , 75 , 76 , 373 , 374 , 411 , 623 , 661	

<code>\glsclosebrace</code>	58,	<code>\glsdescwidth</code>	277, 277, 278–
	176, 180, 181, 402, 468, 471,		280, 282, 283, 285–288, 288,
	573, 574, 652, 720, 723, 826, 827		289–293, 300, 300, 301–303,
<code>\glscompositor</code>	46,		305–307, 307, 308–313, 595–
	46, 47, 57, 183, 395, 401, 473,		598, 600–604, 608–615, 849,
	573, 575, 645, 651, 724, 826, 828		850, 852–858, 861–863, 865–869
<code>\glscounter</code>	20,	<code>\glsdetoklabel</code>	
	20, 41, 53, 71, 96, 122, 380,		. . . 33, 63, 63, 64–68, 76, 77,
	392, 398, 409, 424, 441, 570,		84, 89–93, 99, 104–108, 121,
	630, 642, 648, 659, 675, 692, 823		163, 165, 173, 174, 194–196,
<code>\glscurrententrylabel</code>			203, 205, 213, 216, 217, 221,
 209, 213, 489, 490, 741, 742		222, 224, 227, 229, 261–266,
<code>\glscurrentfieldvalue</code>			319, 325–327, 341–344, 369,
	. . . 67, 68, 68, 407, 408, 657, 658		370, 404–407, 412, 415, 419–
<code>\glscustomtext</code>	112, 114, 116,		421, 426, 428–430, 440, 462,
	134–139, 155–162, 251, 252,		463, 466, 467, 480–482, 486,
	345, 347, 349–352, 362, 363,		490, 492, 493, 496–498, 500,
	432, 435–437, 449–452, 458–		501, 532–535, 541–543, 568,
	462, 514, 519, 520, 523, 525,		569, 587, 619, 638, 655–657,
	527, 543, 546, 548–551, 559,		662, 666, 670–672, 676, 679–
	683, 685–688, 701–703, 709–		681, 691, 714, 718, 719, 732–
	714, 766, 772, 775–778, 780,		734, 737, 738, 742, 744, 745,
	796, 799, 801, 803–805, 812, 813		748–750, 752, 753, 785–788,
<code>\GlsDeclareNoHyperList</code>			794–796, 821, 822, 840, 841, 872
 20, 21, 380, 630	<code>\glsdisablehyper</code>	133, 449, 700
<code>\glsdefaultshortaccess</code>		<code>\Glsdisp</code>	139, 139
 368, 368, 564–567, 817–820	<code>\glsdisp</code>	138, 139, 452, 703
<code>\glsdefaulttype</code>	17, 17,	<code>\glsdisplay</code> 117, 432, 438, 439, 683, 689	
	18, 31, 48, 60, 62, 69, 70, 94,	<code>\glsdisplayfirst</code>	
	111, 198, 210, 211, 378, 396,	 117, 432, 438, 683, 689
	403, 404, 408, 422, 432, 438,	<code>\glsdisplaynumberlist</code>	
	483, 489, 490, 628, 646, 653,	 172, 195, 466, 481, 718, 733
	654, 658, 673, 683, 689, 735, 741	<code>\glsdohyperlink</code>	
<code>\glsdefmain</code> 132, 133, 448, 449, 700, 701
	. . . 17, 18, 72, 378, 409, 628, 659	<code>\glsdohypertarget</code>	
<code>\glsdefs@newdocentry</code> . . .	83, 84, 666	 132, 133, 448, 449, 700, 701
<code>\GLSdesc</code>	145, 454, 706	<code>\glsdoifexists</code>	63,
<code>\Glsdesc</code>	145, 145, 454, 706		65, 67, 89–92, 103, 104, 134–
<code>\glsdesc</code>	144, 145, 454, 706		139, 155–162, 168, 172–174,
<code>\GLSdescplural</code>	146, 455, 706		195, 196, 327–331, 350–352,
<code>\Glsdescplural</code> . . .	146, 146, 454, 706		357, 358, 405–407, 419–421,
<code>\glsdescplural</code> . . .	145, 146, 454, 706		427, 428, 449–452, 458–462,
<code>\glsdescriptionaccessdisplay</code> . .			464, 466, 467, 481, 482, 550,
 342, 354,		551, 555, 588–590, 655–657,
	358, 541, 544–546, 553, 555,		670–672, 678, 679, 701–704,
	556, 795, 798, 799, 806, 808, 809		709–713, 716, 718, 719, 733,
<code>\glsdescriptionpluralaccessdisplay</code>			734, 803, 804, 808, 809, 841–844
 343, 354,	<code>\glsdoifexistsordo</code>	64, 120,
	541, 543, 544, 553, 795–797, 806		163, 405, 440, 462, 655, 690, 714

<code>\glsdofexistsorwarn</code>	431, 433, 434, 436, 453, 464, 545, 547, 548, 552, 682, 684, 685, 687, 704, 716, 798–801, 805
. 64 , 223, 229, 230, 405, 498, 501, 502, 655, 750, 753, 754	
<code>\glsdofnoexists</code> 64 , 81, 92, 405, 414, 421, 655, 664, 672	
<code>\glsdonohyperlink</code> 122, 132, 132 , 133, 441, 448, 449, 692, 700	
<code>\glsdoparenifnotempty</code> . 259 , 527, 779	
<code>\glsdosanitizesort</code> . 14, 376, 377, 626	
<code>\glsdoshowtarget</code> 6 , 132	
<code>\glsenableentrycount</code> 106 , 429, 430, 680, 681	
<code>\glsenablehyper</code> 133 , 449, 700	
<code>\glscapwrcontent</code> 122 , 122	
<code>\glsentryaccess</code> 337 , 342, 539, 541, 792, 794	
<code>\glsentrycounter</code> 122 , 233, 236, 441, 503, 505, 692, 755, 757	
<code>\glsentrycounterfalse</code> . 12, 376, 626	
<code>\glsentrycounterlabel</code> 228 , 228 , 500, 501, 752, 753	
<code>\GlsEntryCounterLabelPrefix</code> 225 , 227, 499, 500, 751, 752	
<code>\glsentrycountertrue</code> . . 12, 376, 626	
<code>\glsentrycurrcount</code> 106 , 108, 429–431, 680–682	
<code>\Glsentrydesc</code> 145 , 166, 166 , 230, 354, 358, 454, 463, 502, 553, 555, 706, 715, 754, 806, 808	
<code>\glsentrydesc</code> 145 , 166, 166 , 230, 354, 358, 433–435, 454, 463, 502, 544–546, 553, 555, 684–686, 706, 715, 753, 798, 799, 806, 808	
<code>\glsentrydescaccess</code> 338 , 343, 539, 541, 792, 795	
<code>\Glsentrydescplural</code> 146 , 166, 166 , 354, 455, 464, 553, 706, 715, 806	
<code>\glsentrydescplural</code> 146 , 166, 166 , 354, 432, 433, 454, 455, 463, 543, 544, 553, 683, 684, 706, 715, 796, 797, 806	
<code>\glsentrydescpluralaccess</code> 338 , 343, 539, 542, 792, 795	
<code>\Glsentryfirst</code> 110, 113, 141, 167 , 168, 346, 353, 431, 434, 436, 453, 464, 545, 547, 552, 682, 685, 687, 704, 716, 798, 801, 805	
<code>\glsentryfirst</code> . . . 109, 113, 114, 141, 167 , 168, 346, 347, 353,	431, 433, 434, 436, 453, 464, 545, 547, 548, 552, 682, 684, 685, 687, 704, 716, 798–801, 805
<code>\glsentryfirstaccess</code> 338 , 342, 539, 541, 792, 794	
<code>\Glsentryfirstplural</code> 111, 112, 143, 168, 168 , 345, 353, 432, 433, 435, 453, 464, 544, 547, 552, 683, 684, 686, 705, 716, 797, 800, 805	
<code>\glsentryfirstplural</code> 110, 112, 113, 143, 168, 168 , 345, 346, 353, 431–433, 435, 453, 454, 464, 543, 544, 546, 547, 552, 682–684, 686, 705, 716, 796, 797, 799, 800, 805, 806	
<code>\glsentryfirstpluralaccess</code> 338 , 342, 539, 541, 792, 794	
<code>\glsentryfmt</code> 71, 72, 112 , 409, 432, 659, 660, 683	
<code>\Glsentryfull</code> 172 , 172 , 244, 253, 254, 364, 366, 466, 509, 515, 516, 560, 562, 718, 761, 767, 768, 813, 815	
<code>\glsentryfull</code> 172 , 172 , 244, 253, 254, 364, 366, 466, 509, 515, 516, 560, 562, 717, 761, 767, 768, 813, 815	
<code>\Glsentryfullpl</code> 172 , 172 , 244, 253, 254, 364, 366, 466, 509, 515, 517, 560, 562, 718, 761, 767, 768, 814, 815	
<code>\glsentryfullpl</code> 172 , 172 , 244, 253, 254, 364, 366, 466, 509, 515, 517, 560, 562, 718, 761, 767, 768, 813, 815	
<code>\glsentryitem</code> 228 , 271, 273–279, 281, 289, 290, 292, 300, 302, 304, 307, 309, 311, 315, 316, 318, 320, 501, 576–581, 583–585, 592, 594–597, 601–603, 608–610, 612–614, 616–619, 753, 829– 838, 845, 847–851, 855–857, 862–864, 866–868, 870, 871, 873	
<code>\Glsentrylong</code> 110, 159, 165, 171, 171 , 172, 246, 247, 252, 253, 350, 352, 359, 363, 364, 431, 460, 463, 466, 510, 514, 515, 550, 551, 556, 559, 560,	

682, 712, 714, 717, 718, 762, 766, 767, 803, 804, 809, 812, 813	\Glsentryprefix 326, <u>326</u> , 329, 587, 589, 840, 842
\glsentrylong . 109, 116, 159, 160, 165, 171, <u>171</u> , 172, 246–256, 349, 351, 352, 359–368, 431, 438, 460, 461, 463, 466, 510– 518, 525, 549, 551, 556–563, 682, 688, 712, 714, 717, 718, 762–770, 778, 803–805, 809–817	\glsentryprefix . . . <u>325</u> , 326, 327, 330, 587, 588, 590, 840, 841, 843
\glsentrylongaccess 339, 343, 539, 542, 792, 795	\Glsentryprefixfirst <u>325</u> , 326, 329, 587, 589, 840, 842
\Glsentrylongpl 111, 161, 171, <u>171</u> , 172, 246, 247, 251–253, 350, 359, 362, 364, 432, 462, 466, 510, 511, 514, 515, 550, 556, 559, 560, 683, 713, 717, 718, 762, 763, 766, 767, 803, 809, 812–814	\glsentryprefixfirst 325, 326, 327, 330, 587, 588, 590, 840, 841, 843
\glsentrylongpl 110, 116, 161, 162, 171, <u>171</u> , 172, 246–248, 251–254, 350, 359, 360, 362, 364–366, 431, 438, 461, 462, 466, 510, 511, 514– 517, 525, 531, 550, 556, 557, 559–562, 682, 689, 713, 714, 717, 718, 762, 763, 766–768, 778, 784, 803, 809, 810, 812–815	\Glsentryprefixfirstplural 326, <u>326</u> , 330, 587, 590, 840, 843
\glsentrylongpluralaccess 339, 343, 539, 542, 793, 795	\glsentryprefixfirstplural <u>325</u> , 326, 328, 331, 587, 588, 591, 840, 842, 844
\Glsentryname 144, 164, <u>164</u> , 229, 354, 358, 454, 463, 501, 553, 555, 705, 714, 753, 806, 808	\Glsentryprefixplural 326, <u>326</u> , 329, 587, 589, 840, 843
\glsentryname 144, 164, <u>164</u> , 209, 319, 354, 357, 454, 463, 552, 553, 555, 619, 705, 706, 714, 806, 808, 872	\glsentryprefixplural <u>325</u> , 326, 328, 331, 587, 588, 590, 840, 842, 844
\glsentrynumberlist 172, <u>172</u> , 195, 466, 481, 718, 733	\glsentryprevcount 106–108, 429, 430, 680, 681
\glsentryparent <u>169</u> , 465, 716	\Glsentryshort . . . 115, 155, 165, 171, <u>171</u> , 248, 254, 348, 350, 360, 365, 366, 437, 458, 463, 465, 511, 516, 549, 550, 557, 561, 562, 688, 710, 715, 717, 763, 768, 802, 803, 810, 814, 815
\Glsentryplural 112, 142, 167, <u>167</u> , 345, 353, 433, 435, 453, 464, 543, 546, 552, 683, 686, 705, 715, 797, 800, 805	\glsentryshort 115, 116, 155, 156, 165, <u>170</u> , 171, 172, 245–256, 348–351, 359–368, 437, 438, 458, 459, 463, 465, 466, 509–518, 548–551, 556– 563, 688, 709, 710, 715, 717, 718, 761–770, 802–804, 809–817
\glsentryplural . . 112, 113, 142, <u>166</u> , 167, 345, 346, 353, 432, 433, 435, 453, 464, 543, 544, 546, 547, 552, 683, 684, 686, 705, 715, 796, 797, 799, 800, 805	\glsentryshortaccess 338, 343, 539, 542, 792, 795
\glsentrypluralaccess 338, 342, 539, 541, 792, 794	\Glsentryshortpl 114, 157, 171, <u>171</u> , 248, 254, 348, 360, 365, 366, 436, 459, 466, 511, 516, 517, 548, 557, 561, 562, 687, 711, 717, 763, 768, 801, 810, 814, 815
	\glsentryshortpl 114, 116, 157, 158, 171, <u>171</u> , 172, 246–248, 252–254, 347, 348, 350, 359, 360, 364–366, 436, 438, 459, 460, 466, 510, 511, 515–517, 531, 548, 550, 556, 557, 560–562, 687, 689, 711, 717, 718, 762, 763, 767, 768, 784, 801, 803, 809, 810, 813–815

<code>\glentryshortpluralaccess</code>	<code>\glentryuseriiaccess</code>
. . . . 338 , 343 , 539 , 542 , 792 , 795 339 , 344 , 539 , 542 , 793 , 795
<code>\glentrysort</code> 169 , 465 , 716	<code>\Glentryuseriii</code> . . 151 , 169 , 170 ,
<code>\Glentrysymbol</code> . . 147 , 167 , 167 ,	356 , 456 , 465 , 554 , 708 , 717 , 807
230 , 355 , 358 , 455 , 464 , 502 ,	<code>\glentryuseriii</code> . 151 , 169 , 170 ,
553 , 556 , 707 , 715 , 754 , 806 , 809	356 , 456 , 465 , 554 , 708 , 717 , 807
<code>\glentrysymbol</code>	<code>\glentryuseriiiaccess</code>
147 , 167 , 167 , 230 , 355 , 358 , 339 , 344 , 540 , 542 , 793 , 795
433 – 435 , 455 , 464 , 502 , 520 ,	<code>\Glentryuseriv</code> . . 152 , 170 , 170 ,
523 , 527 , 544 – 546 , 553 , 555 ,	356 , 457 , 465 , 554 , 708 , 717 , 807
684 – 686 , 706 , 707 , 715 , 754 ,	<code>\glentryuseriv</code> 152 , 170 , 170 , 356 ,
772 , 776 , 780 , 798 , 799 , 806 , 809	457 , 465 , 554 , 708 , 717 , 807 , 808
<code>\glentrysymbolaccess</code>	<code>\glentryuserivaccess</code>
. . . . 338 , 342 , 539 , 541 , 792 , 794 339 , 344 , 540 , 542 , 793 , 796
<code>\Glentrysymbolplural</code>	<code>\Glentryuserv</code> . . . 153 , 170 , 170 ,
. 148 , 167 , 167 ,	357 , 457 , 465 , 554 , 709 , 717 , 808
355 , 455 , 464 , 553 , 707 , 716 , 807	<code>\glentryuserv</code> 153 , 170 , 170 , 357 ,
<code>\glentrysymbolplural</code>	457 , 465 , 554 , 555 , 709 , 717 , 808
. . . . 148 , 167 , 167 , 355 , 432 ,	<code>\glentryuservaccess</code>
433 , 455 , 464 , 520 , 523 , 527 , 339 , 344 , 540 , 542 , 793 , 796
543 , 544 , 553 , 683 , 684 , 707 ,	<code>\Glentryuservi</code> . . 154 , 170 , 170 ,
716 , 772 , 776 , 780 , 796 , 797 , 807	357 , 457 , 465 , 555 , 709 , 717 , 808
<code>\glentrysymbolpluralaccess</code> . . .	<code>\glentryuservi</code> 154 , 170 , 170 , 357 ,
. . . . 338 , 342 , 539 , 541 , 792 , 795	457 , 458 , 465 , 555 , 709 , 717 , 808
<code>\Glentrytext</code> 113 ,	<code>\glentryuserviaccess</code>
140 , 166 , 166 , 346 , 352 , 434 , 339 , 344 , 540 , 543 , 793 , 796
436 , 452 , 464 , 545 , 547 , 552 ,	<code>\glscsclocationsfalse</code>
685 , 687 , 704 , 715 , 798 , 800 , 805 10 , 194 , 480 , 732
<code>\glentrytext</code> 113 ,	<code>\glscsclocationstrue</code> 375 , 624
114 , 140 , 166 , 166 , 174 , 209 ,	<code>\glsexpandfields</code> 81 , 414 , 664
346 , 347 , 352 , 433 , 434 , 436 ,	<code>\glsfieldaccsupp</code>
452 , 464 , 467 , 489 , 544 , 545 ,	340 , 340 , 341 , 540 , 541 , 793 , 794
547 , 548 , 552 , 684 – 687 , 704 ,	<code>\glsfielddef</code> 90 , 420 , 670
715 , 719 , 740 , 798 , 800 , 801 , 805	<code>\glsfielddedef</code> 89 , 419 , 670
<code>\glentrytextaccess</code>	<code>\glsfieldfetch</code>
. . . . 337 , 342 , 539 , 541 , 792 , 794 90 , 168 , 420 , 464 , 671 , 716
<code>\glentrytitlecase</code> 168 , 464 , 465 , 716	<code>\glsfieldgdef</code> 89 , 419 , 670
<code>\glentrytype</code>	<code>\glsfieldxdef</code> 89 , 419 , 670
. . . . 94 , 168 , 422 , 465 , 673 , 716	<code>\glsfindwidesttoplevelname</code>
<code>\Glentryuseri</code> . . . 149 , 169 , 169 , 319 , 618 , 872
355 , 456 , 465 , 554 , 707 , 717 , 807	<code>\GLSfirst</code> 141 , 453 , 704
<code>\glentryuseri</code> 149 , 169 , 169 , 355 ,	<code>\Glsfirst</code> 141 , 141 , 453 , 704
455 , 456 , 465 , 554 , 707 , 716 , 807	<code>\glsfirst</code> 141 , 141 , 453 , 704
<code>\glentryuseriaccess</code>	<code>\glsfirstaccessdisplay</code>
. . . . 339 , 343 , 539 , 542 , 793 , 795	342 , 346 , 347 , 353 , 541 , 545 ,
<code>\Glentryuserii</code> . . 150 , 169 , 169 ,	547 , 548 , 552 , 794 , 798 – 801 , 805
356 , 456 , 465 , 554 , 708 , 717 , 807	<code>\GLSfirstplural</code> 143 , 453 , 705
<code>\glentryuserii</code> 150 , 169 , 169 , 356 ,	<code>\Glsfirstplural</code> . . 143 , 143 , 453 , 705
456 , 465 , 554 , 707 , 708 , 717 , 807	<code>\glsfirstplural</code> . . 142 , 143 , 453 , 705

<code>\glsfirstpluralaccessdisplay</code> ..	<code>\glshypernavsep</code> ..
..... 342, 345, 353,	270, 270, 592, 845
541, 543, 544, 546, 547, 552,	<code>\glshypernumber</code>
794, 796, 797, 799, 800, 805, 806	48, 235,
<code>\glsfirstpluralacessdisplay</code> ...	237, 395, 504–506, 646, 756–758
..... 346, 547, 800	<code>\glsifhyper</code>
<code>\glsgenacfmt</code> ..	119, 439, 690
114, 246, 247, 253,	<code>\glsifhyperon</code> .
347, 359, 365, 436, 510, 511,	121, 439, 440, 690, 691
516, 548, 556, 557, 561, 687,	<code>\glsIfListOfAcronyms</code>
762, 763, 767, 801, 809, 810, 814 19, 19, 20, 379, 380, 629
<code>\glsgentryfmt</code> 112, 112, 246, 247,	<code>\glsifmeasuring</code>
252, 253, 258, 345, 359, 363,	102, 102
365, 435, 510, 511, 514, 516,	<code>\glsifplural</code> 112, 114, 115, 134–139,
518, 520, 521, 523, 525, 527,	155–162, 251, 345, 347, 348,
529, 531, 546, 556, 557, 559,	350–352, 362, 432, 435–437,
561, 686, 762, 763, 766, 767,	449–452, 458–462, 514, 520,
770, 772, 774, 775, 778, 780,	523, 525, 527, 543, 546, 548–
782, 784, 799, 809, 810, 813, 814	551, 559, 683, 686–688, 701–
<code>\glsgetgrouplabel</code>	703, 709–714, 766, 772, 775,
232, 503, 755	778, 780, 796, 799, 801–805, 812
<code>\glsgetgrouptitle</code>	<code>\glsifusedtranslatordict</code> 27, 385, 635
232, 270,	<code>\glsifusetranslator</code> 27–29, 44, 45,
274–276, 294–299, 315–319,	323, 385, 386, 393, 394, 569,
322, 503, 592, 594, 595, 605–	586, 634–636, 643, 644, 822, 839
608, 616–618, 620, 755, 845,	<code>\glsgignore</code>
847, 848, 858–861, 870–872, 874	175, 468, 719
<code>\glsglossarymark</code> ...	<code>\glsindexingsetting</code> .
49, 49, 396, 646	13, 16, 191, 193
<code>\glsgroupheading</code>	<code>\glsindexonlyfirstfalse</code> 29, 386, 636
..... 181, 222, 231, 271, 274–	<code>\glsinlinedescformat</code>
276, 278, 279, 281, 289, 290, 271, 272, 576, 593, 829, 846
292, 294–300, 302, 304, 307,	<code>\glsinlinedopostchild</code>
309, 311, 314–320, 322, 471,	271,
497, 503, 574, 592, 594–597,	272, 576, 592, 593, 829, 845, 846
601–603, 605–610, 612–614,	<code>\glsinlineemptydescformat</code>
616–620, 723, 749, 755, 827,	271, 272, 576, 592, 593, 829, 846
845, 847–851, 855–864, 866–874	<code>\glsinlineifhaschildren</code> ..
<code>\glsgroupskip</code>	271, 272
..... 180, 181, 222, 231, 272, 274,	<code>\glsinlinenameformat</code>
278, 279, 281, 284, 285, 289,	271, 272, 576, 592, 593, 829, 846
291, 292, 301, 302, 304, 308,	<code>\glsinlineparentchildseparator</code>
310, 312, 315, 317, 318, 321, 271, 272, 576, 593, 829, 846
471, 497, 503, 573, 574, 593,	<code>\glsinlinepostchild</code>
594, 596, 597, 599, 602, 603, 271, 272, 576, 593, 829, 846
608–610, 612–614, 616–618,	<code>\glsinlineseparator</code>
620, 723, 749, 755, 826, 827, 271, 272, 576, 593, 829, 846
846, 847, 849–853, 855–857,	<code>\glsinlinesubdescformat</code>
862–864, 866–868, 870–872, 874	272, 273, 576, 593, 829, 846, 847
<code>\glshyperfirstfalse</code>	<code>\glsinlinesubnameformat</code>
..... 253, 365, 516, 561, 768, 814	272, 273, 576, 593, 829, 846, 847
<code>\glshyperfirsttrue</code>	<code>\glsinlinesubseparator</code>
29, 386, 636 272, 272, 576, 593, 829, 846
<code>\glshyperlink</code>	<code>\glsinsert</code> ...
..... 173, 209, 467, 489, 719, 740	112–116, 134–139,
	155–162, 251, 252, 345–352,
	362, 363, 432–437, 449–452,
	458–462, 514, 520, 523, 525,
	527, 543–551, 559, 683–688,

701–703, 709–714, 766, 772, 775–778, 780, 796–805, 812, 813	<code>\glslongpluralaccessdisplay</code> 343, 350, 359, 360, 362, 364–366, 542, 550, 556, 557, 559–562, 564, 795, 803, 809, 810, 812–815, 817
<code>\glskeylisttok</code> 243, 243, 244, 368, 508, 519–522, 524, 526, 528– 532, 564–568, 760, 770, 771, 773–779, 781–783, 785, 817–821	<code>\glslongpluralaccesskey</code> 369, 568, 821
<code>\glslabel</code> .. 93, 112–116, 120–122, 155–162, 246, 247, 251–253, 345–349, 359, 362, 363, 365, 422, 432–437, 440, 441, 458– 462, 510, 511, 514, 516, 520, 523, 525, 527, 543–549, 556, 557, 559, 561, 672, 683–688, 691, 692, 710–714, 762, 763, 766, 767, 772, 775, 776, 778, 780, 796–802, 809, 810, 812, 814	<code>\glslongpluralkey</code> 239, 369, 506, 568, 758, 821
<code>\glslabeltok</code> 243, 243, 244, 508, 518–532, 564–567, 760, 770–780, 782–785, 817–820	<code>\glslongtok</code> 243, 243, 244, 246–248, 252, 253, 359, 363, 365, 368, 508–511, 515, 516, 518–526, 528–532, 556, 557, 560, 561, 564–567, 760– 763, 766, 768, 770–779, 781– 785, 809, 810, 813, 814, 817–821
<code>\glsletentryfield</code> 163, 462, 714	<code>\glslowercase</code> 2
<code>\Glslink</code> 120, 120	<code>\glsLTpenaltycheck</code> 287, 287, 288, 600, 601, 854
<code>\glslink</code> .. 120, 120, 244, 252–254, 258, 363–366, 439, 508, 509, 515, 516, 519, 560–562, 690, 760, 761, 766–768, 772, 813, 815	<code>\glsmakefirstuc</code> 39
<code>\glslinkcheckfirsthyperhook</code> 121, 121, 440, 691	<code>\glsmcols</code> 294, 294, 295–299, 604–607, 858–861
<code>\glslinkpostsetkeys</code> 121, 122, 440, 441, 691	<code>\glsmeasuredepth</code> 102
<code>\glslinkvar</code> 119, 119, 439, 690	<code>\glsmeasureheight</code> 102, 132
<code>\glslistdottedwidth</code> 276, 276, 577, 595, 830, 848	<code>\glsmeasurewidth</code> 103, 319–321
<code>\glslistexpandedname</code> .. 273, 274, 847	<code>\glsmfuaddmap</code> 3, 88, 109, 110, 120, 134, 137, 139–155, 157, 159, 161, 164, 166–172, 229, 230, 241, 242, 326, 328, 329
<code>\glslistgroupheaderfmt</code> 273, 274–276, 594, 595, 847, 848	<code>\glsmfublocker</code> 3, 3, 88, 135, 138, 140–154, 156, 158, 160, 162
<code>\glslistinit</code> 273, 274, 847	<code>\glsmfuexcl</code> 3, 3, 174
<code>\glslistnavigationitem</code> 273, 275, 276, 594, 847, 848	<code>\glsmoveentry</code> 99, 426, 676
<code>\glslocalreset</code> 103, 105, 427, 428, 678, 679	<code>\GLSname</code> 144, 454, 705
<code>\glslocalresetall</code> 105, 428, 679	<code>\Glsname</code> 144, 144, 454, 705
<code>\glslocalunset</code> 104, 105, 118, 428, 429, 449–452, 679, 680, 701–703	<code>\glsname</code> 143, 144, 454, 705
<code>\glslocalunsetall</code> 105, 429, 680	<code>\glsnameaccessdisplay</code> 341, 354, 357, 358, 541, 552, 553, 555, 556, 794, 806, 808, 809
<code>\glslocationcstoencap</code> 48	<code>\glsnamefont</code> 229, 234, 357, 358, 501, 502, 504, 535, 555, 753, 754, 756, 788, 808
<code>\glslongaccessdisplay</code> . 343, 349– 352, 359–368, 542, 549–551, 556–564, 795, 803–805, 809–817	<code>\glsnavhyperlink</code> 268, 270, 591, 592, 844, 845
<code>\glslongaccesskey</code> 368, 568, 821	<code>\glsnavhyperlinkname</code> 268, 268, 269, 591, 844
<code>\glslongkey</code> 239, 368, 506, 568, 758, 821	<code>\glsnavhypertarget</code> 268, 274–276, 295–299, 316, 317, 319, 322, 591, 594, 595, 605–608, 617, 618, 620, 844, 847, 848, 858–861, 870–872, 874

<code>\glsnavigation</code>	269, 274–276, 295–299, 316–318, 322, 591, 594, 605–607, 617, 618, 620, 845, 847, 848, 858–861, 870–872, 874	<code>\glsorg@theglossary</code>	4, 4, 371, 621
<code>\glsnextpages</code>	10, 76, 212, 227, 375, 411, 490, 500, 624, 661, 742, 752	<code>\glspagelistwidth</code>	277, 279, 280, 282, 283, 285–287, 288, 290–293, 300, 302, 303, 305, 306, 307, 309–313, 595–598, 600–604, 608–615, 849, 850, 852–858, 862, 863, 865–869
<code>\glsnoexpandfields</code>	81, 414, 664	<code>\glspar</code>	45, 395, 645
<code>\glsnogroupskipfalse</code>	12, 375, 625	<code>\glspatchLtoutput</code>	284–286, 287, 599, 600, 852–854
<code>\glsnoidxdisplayloc</code>	196, 197, 223, 482, 483, 498, 734, 735, 750	<code>\glspatchtabularx</code>	103, 427, 678
<code>\glsnoidxdisplayloclisthandler</code>	196, 223, 482, 497, 734, 749	<code>\glspenaltygroupskip</code>	284, 285, 287, 599, 600, 852–854
<code>\glsnoidxloclist</code>	195, 222, 222, 481, 497, 733, 748, 749	<code>\glsspercentchar</code>	84, 176, 180, 182, 415, 416, 468, 471, 472, 666, 667, 720, 722, 724
<code>\glsnoidxloclisthandler</code>	222, 223, 497, 749	<code>\GLSpl</code>	138, 451, 703
<code>\glsnoidxnumberlistloophandler</code>	196, 197, 482, 483, 734	<code>\Glspl</code>	111, 137, 137, 257, 432, 451, 518, 682, 702, 770
<code>\glsnoidxprenumberlist</code>	76, 221, 222	<code>\glspl</code>	110, 136, 137, 257, 431, 450, 518, 682, 702, 770
<code>\glsnoidxstripaccents</code>	24, 24, 382, 383, 632, 633	<code>\GLSplural</code>	142, 453, 705
<code>\glsnomakeindexwarning</code>	52, 183, 398, 473, 648, 725	<code>\Glsplural</code>	142, 142, 453, 705
<code>\glsnonextpages</code>	76, 212, 226, 411, 490, 500, 661, 742, 752	<code>\glsplural</code>	141, 142, 453, 705
<code>\glsnopostdotfalse</code>	11, 375, 625	<code>\glspluralaccessdisplay</code>	342, 345, 346, 353, 541, 543, 544, 546, 547, 552, 794, 796, 797, 799, 800, 805
<code>\glsnoxindywarning</code>	47, 52, 53, 55, 56, 58–61, 176, 177, 395, 398–400, 402, 403, 468, 469, 645, 648–650, 652–654, 720, 721	<code>\glspluralsuffix</code>	43, 43, 96, 244, 246–248, 359, 360, 364–367, 393, 424, 510, 511, 556, 557, 561, 563, 643, 674, 675, 762, 763, 810, 814–816
<code>\glsnumberformat</code>	47, 173, 395, 466, 467, 646, 718	<code>\glspostdescription</code>	11, 45, 272, 274, 275, 278, 289, 300, 307, 308, 315, 316, 318, 320, 321, 375, 395, 576–584, 593–596, 601, 608, 612, 616–620, 625, 645, 830, 832–838, 846–849, 855, 862, 866, 870–873
<code>\glsnumberlistloop</code>	196, 197, 482, 734	<code>\glspostinline</code>	271, 272, 592, 593, 845, 846
<code>\glsnumbersgroupname</code>	38, 42, 45, 233, 391, 393, 394, 503, 641, 643, 645, 755	<code>\glspostlinkhook</code>	120, 120, 134–139, 155–162, 350–352, 440, 449–452, 458–462, 550–552, 691, 701–704, 710–714, 803–805
<code>\glsnumlistlastsep</code>	173, 173, 195, 467, 482, 719, 733	<code>\glsprefixsep</code>	327, 327, 328–331, 588–591, 841–844
<code>\glsnumlistparser</code>	173, 192, 467, 480, 719, 731	<code>\glsprestandardsort</code>	13, 14, 376, 377, 626
<code>\glsnumlistsep</code>	173, 173, 195, 467, 482, 719, 733		
<code>\glsopenbrace</code>	58, 176, 180, 181, 402, 468, 471, 573, 574, 652, 719, 723, 826, 827		
<code>\glsorder</code>	30, 30, 188–190, 192, 218, 387, 477–479, 494, 636, 637, 728–731, 746		
<code>\glsorg@endtheglossary</code>	4, 4, 371, 621		
<code>\glsorg@PrintChanges</code>	4, 371, 621		

<code>\glsquote</code>	176 , 468 , 720	<code>\GlsSetXdyNumberGroupOrder</code>	
<code>\glsrefentry</code>	227 , 500 , 752		177 , 468 , 469 , 720
<code>\glsreset</code> 103 , 105 , 427 , 428 , 678 , 679		<code>\GlsSetXdyStyles</code>	59 , 403 , 653
<code>\glsresetall</code>	105 , 428 , 679	<code>\glsshortaccessdisplay</code>	
<code>\glsresetcurrcountfalse</code>	106		165 , 343 , 348–351 , 359–
<code>\glsresetentrycounter</code>	227 , 500 , 752		368 , 463 , 542 , 548–551 , 556–
<code>\glsresetentrylist</code>	180 , 220 , 226 , 226 , 471 , 496 , 499 , 500 , 573 , 723 , 747 , 751 , 752 , 826		564 , 715 , 795 , 802–804 , 809–817
<code>\glsresetsubentrycounter</code>		<code>\glsshortaccesskey</code>	368 , 568 , 821
	227 , 228 , 271 , 500 , 501 , 576 , 593 , 752 , 753 , 829 , 846	<code>\glsshortaccsupp</code>	341 , 341 , 540 , 793
<code>\glsrestoreLToutput</code>	287 , 600 , 854	<code>\glsshortkey</code>	239 , 368 , 506 , 568 , 758 , 821
<code>\glssanitizesortfalse</code>	26 , 384 , 634	<code>\glsshortplaccsupp</code>	341 , 540 , 793
<code>\glssanitizesorttrue</code> 26 , 384 , 633 , 634		<code>\glsshortpluralaccessdisplay</code>	343 , 347 , 348 , 350 , 359 , 360 , 364–366 , 542 , 548 , 550 , 556 , 557 , 560–562 , 564 , 795 , 801 , 803 , 809 , 810 , 813–815 , 817
<code>\glssavenumberlistfalse</code> 10 , 375 , 624		<code>\glsshortpluralaccesskey</code>	368 , 568 , 821
<code>\glssavewritesfalse</code>	36 , 389 , 639	<code>\glsshortpluralkey</code>	239 , 368 , 506 , 568 , 758 , 821
<code>\glssee</code>	208 , 488 , 740	<code>\glsshorttok</code>	243 , 243 , 244 , 368 , 508 , 518–526 , 528–532 , 564–568 , 760 , 770–785 , 817–821
<code>\glsseeformat</code>	179 , 194 , 196 , 208 , 470 , 481 , 482 , 488 , 573 , 722 , 732 , 734 , 740 , 826	<code>\glsshowaccsupp</code> 5 , 7 , 372 , 373 , 622 , 623	
<code>\glsseeitem</code>	209 , 209 , 488 , 489 , 740	<code>\glsshowtarget</code>	5 , 6 , 372 , 373 , 622
<code>\glsseeitemformat</code> 209 , 209 , 489 , 740		<code>\glsshowtargetfont</code> 6 , 6 , 372 , 373 , 622	
<code>\glsseelastsep</code>	209 , 209 , 488 , 740	<code>\glsshowtargetfonttext</code>	6 , 6
<code>\glsseelist</code>	208 , 208 , 488 , 740	<code>\glsshowtargetinner</code>	6 , 6
<code>\glsseesep</code>	209 , 209 , 488 , 740	<code>\glsshowtargetouter</code>	6 , 6 , 373 , 622
<code>\glssentencecase</code>		<code>\glsshowtargetsymbols</code>	6 , 6 , 373 , 622
	2 , 116 , 117 , 120 , 139 , 163–165	<code>\glssortnumberfmt</code>	14 , 15 , 16 , 377 , 378 , 626 , 627
<code>\glsSetAlphaCompositor</code> 47 , 395 , 645		<code>\glsspace</code>	240 , 240 , 507 , 758 , 771
<code>\glsSetCompositor</code>	46 , 395 , 645	<code>\glsstepentry</code>	227 , 228 , 500 , 501 , 752 , 753
<code>\glssetexpandfield</code>	22 , 22 , 25–27 , 381 , 383 , 384 , 631 , 633 , 634	<code>\glsstepsubentry</code>	227 , 228 , 500 , 501 , 752 , 753
<code>\glssetnoexpandfield</code>	22 , 22 , 23 , 25 , 26 , 381 , 383 , 384 , 631 , 633 , 634	<code>\glssubentrycounterfalse</code> 13 , 376 , 626	
<code>\GlsSetQuote</code>	93 , 182 , 183 , 421 , 472 , 473 , 476 , 672 , 723 , 725 , 728	<code>\glssubentrycounterlabel</code>	228 , 228 , 500 , 501 , 752 , 753
<code>\glsSetSuffixF</code>	47 , 395 , 645	<code>\glssubentryitem</code> 228 , 272 , 274–276 , 278 , 279 , 281 , 289 , 290 , 292 , 300 , 302 , 304 , 308 , 309 , 311 , 315 , 316 , 318 , 320 , 501 , 576– 585 , 593–597 , 601–603 , 608– 610 , 612–614 , 616–619 , 753 , 829–838 , 846–851 , 855–857 , 862–864 , 866–868 , 870 , 871 , 873	
<code>\glsSetSuffixFF</code>	47 , 395 , 645	<code>\GLSsymbol</code>	147 , 455 , 707
<code>\glssettoctitle</code>	42 , 45 , 212 , 393 , 394 , 490 , 643 , 644 , 742		
<code>\glssetwidest</code>	319 , 619 , 872		
<code>\GlsSetWriteIstHook</code>	183 , 473 , 725		
<code>\GlsSetXdyCodePage</code>	60 , 403 , 653 , 654		
<code>\GlsSetXdyFirstLetterAfterDigits</code>	176 , 468 , 720		
<code>\GlsSetXdyLanguage</code>	60 , 403 , 653		
<code>\GlsSetXdyLocationClassOrder</code>	59 , 402 , 652		
<code>\GlsSetXdyMinRangeLength</code>	177 , 469 , 720 , 721		

<code>\Glssymbol</code>	147, 147 , 455, 706	<code>\glstreeitem</code> . .	294, 295, 314, 314 ,
<code>\glssymbol</code>	146 , 147, 455, 706		604, 605, 615, 616, 858, 859, 869
<code>\glssymbolaccessdisplay</code> . .	342 ,	<code>\glstreenamebox</code>	
	355, 358, 541, 544–546, 553,		319 , 320, 321, 619, 620, 872, 873
	555, 556, 794, 798, 799, 806, 809	<code>\glstreenamefmt</code> . .	313, 313 , 315,
<code>\glssymbolnav</code>	270 , 592, 845		316, 318–321, 615–620, 869–873
<code>\GLSsymbolplural</code>	148 , 455, 707	<code>\glstreenavigationfmt</code>	
<code>\Glssymbolplural</code> .	148, 148 , 455, 707	 295–299, 313 , 316–
<code>\glssymbolplural</code> .	147 , 148, 455, 707		318, 322, 605–607, 615, 617,
<code>\glssymbolpluralaccessdisplay</code> .			618, 620, 858–861, 869–872, 874
 342 , 355, 541,	<code>\glstreepredesc</code>	314 ,
	543, 544, 553, 794, 796, 797, 807		315, 316, 318, 616–618, 869–871
<code>\glssymbolsgroupname</code>		<code>\glstreesubitem</code>	
 38, 42 , 45, 233, 390,		294, 314, 314 , 604, 616, 858, 869
	393, 394, 503, 641, 643, 645, 755	<code>\glstreesubsubitem</code>	
<code>\glstarget</code> . . .	102, 229 , 272–279,		294, 314, 314 , 604, 616, 858, 869
	281, 289, 290, 292, 300, 302,	<code>\glstype</code>	120, 121, 134–
	304, 307–309, 311, 315, 316,		139, 155–162, 350–352, 440,
	318, 320, 321, 501–503, 576–		449–452, 458–462, 550, 551,
	585, 593–597, 601–603, 608–		691, 701–703, 709–714, 803–805
	610, 612–614, 616–620, 753–	<code>\glsucmarkfalse</code>	12, 375, 625
	755, 829–838, 846–851, 855–	<code>\glsucmarktrue</code>	12, 375, 625
	857, 862–864, 866–868, 870–873	<code>\glsunexpandedfieldvalue</code>	
<code>\glstexorpdfstring</code> 90 , 274, 671, 847
 163 , 163 , 164, 168, 202	<code>\glsunset</code>	
<code>\GLStext</code>	140 , 452, 704		103 , 105, 107, 108, 118, 427–
<code>\Glstext</code>	140, 140 , 452, 704		430, 449–452, 678–681, 701–704
<code>\glstext</code>	140, 140 , 452, 704	<code>\glsunsetall</code>	105 , 429, 679
<code>\glstextaccessdisplay</code> .	342 , 346,	<code>\glsupacrpluralsuffix</code>	
	347, 352, 541, 544, 545, 547,	 43 , 248, 249, 255, 393,
	548, 552, 794, 798, 800, 801, 805		512, 517, 521, 524, 526, 529,
<code>\glstextformat</code>	112 , 120,		643, 764, 769, 774, 777, 779, 782
	122, 432, 440, 441, 683, 691, 692	<code>\glsupercase</code>	2 , 50, 88, 113–
<code>\glstextup</code>	43, 239 ,		116, 140–154, 156, 158, 160,
	366, 393, 506, 563, 643, 758, 816		162, 244, 251–254, 330, 331,
<code>\glstildechar</code> . .	54, 176 , 180, 181,		346–349, 351–357, 362–364, 366
	399, 468, 471, 649, 720, 722, 723	<code>\GlsUseAcrEntryDispStyle</code>	
<code>\glstranslatefalse</code> 245 , 246 ,
 28, 29, 385, 386, 635		248–251, 253, 255, 256, 360–
<code>\glstranslatetrue</code>			362, 364, 366, 367, 509–515,
 28, 29, 385, 386, 635, 636		517, 557, 558, 561–563, 761–
<code>\glstreechildpredesc</code>			765, 767, 769, 810–812, 814, 816
	314 , 315, 316, 616, 617, 869–871	<code>\GlsUseAcrStyleDefs</code>	
<code>\glstreegroupheaderfmt</code> 245 , 246 , 248–251, 253,
 294–299, 313 ,		255, 256, 360–362, 364, 366–
	315–319, 322, 605–608, 615–		368, 509, 510, 512–515, 517,
	618, 620, 858–861, 869–872, 874		518, 557, 558, 561–563, 761–
<code>\glstreeindent</code>			767, 769, 810–812, 814, 816, 817
	316, 317 , 318, 320, 321, 580–	<code>\GLSuseri</code>	149 , 456, 707
	582, 617–620, 834, 835, 871, 873	<code>\Glsuseri</code>	149 , 149 , 456, 707

<code>\glsuseri</code>	148 , 149 , 455 , 707	H	
<code>\glsuseriaccessdisplay</code>		<code>\H</code>	24 , 383 , 632
.	343 , 355 , 542 , 554 , 795 , 807	<code>\hangindent</code>	298 , 299 ,
<code>\GLSuserii</code>	150 , 456 , 708		314 , 316 , 318 , 320–322 , 580–
<code>\Glsuserii</code>	150 , 150 , 456 , 708		582 , 607 , 608 , 615 , 617–620 ,
<code>\glsuserii</code>	149 , 150 , 456 , 707		833–835 , 861 , 869–871 , 873 , 874
<code>\glsuseriiaccessdisplay</code>		<code>\hbox</code>	103 ,
.	343 , 356 , 542 , 554 , 795 , 807		276 , 427 , 577 , 595 , 678 , 830 , 848
<code>\GLSuseriii</code>	151 , 456 , 708	<code>\hfill</code>	276 , 577 , 595 , 830 , 848
<code>\Glsuseriii</code>	151 , 151 , 456 , 708	<code>\hline</code>	278–280 , 282 , 289–291 ,
<code>\glsuseriii</code>	150 , 151 , 456 , 708		293 , 301–303 , 305 , 306 , 308–
<code>\glsuseriiiaccessdisplay</code>			313 , 596–598 , 602–604 , 608–
.	344 , 356 , 542 , 554 , 795 , 807		615 , 849–851 , 855–858 , 862–869
<code>\GLSuseriv</code>	152 , 457 , 708	<code>\hsize</code>	276 , 277 , 288 ,
<code>\Glsuseriv</code>	152 , 152 , 457 , 708		300 , 307 , 595 , 601 , 608 , 612 ,
<code>\glsuseriv</code>	151 , 152 , 456 , 708		848 , 849 , 855 , 861 , 862 , 865 , 866
<code>\glsuserivaccessdisplay</code>		<code>\hspace</code>	314 , 616 , 869
.	344 , 356 , 542 , 554 , 795 , 807 , 808	<code>\hss</code>	276 , 577 , 595 , 830 , 848
<code>\GLSuserv</code>	153 , 457 , 709	<code>\ht</code>	287 , 600 , 854
<code>\Glsuserv</code>	153 , 153 , 457 , 709	<code>\hyperbf</code>	237 , 506 , 758
<code>\glsuserv</code>	152 , 153 , 457 , 708	<code>\hyperdef</code>	41 , 392 , 642
<code>\glsuservaccessdisplay</code>		<code>\hyperemph</code>	237 , 506 , 758
.	344 , 357 , 542 , 554 , 555 , 796 , 808	<code>hyperfirst</code> (option)	29
<code>\GLSuservi</code>	154 , 457 , 709	<code>\hyperit</code>	237 , 506 , 758
<code>\Glsuservi</code>	154 , 154 , 457 , 709	<code>\hyperlink</code>	118 , 132 ,
<code>\glsuservi</code>	153 , 154 , 457 , 709		236 , 439 , 448 , 505 , 690 , 700 , 757
<code>\glsuserviaccessdisplay</code>		<code>\hypermd</code>	237 , 506 , 758
.	344 , 357 , 542 , 555 , 796 , 808	<code>\hyperm</code>	237 , 506 , 757
<code>\GlsWarnAddProtectedPageFmt</code>	201 , 202	<code>\hypersc</code>	237 , 506 , 758
<code>\glswrallowprimitivemodsfalse</code>	203	<code>\hypersf</code>	237 , 506 , 758
<code>\glswrallowprimitivemodstrue</code>	486 , 738	<code>\hypersl</code>	237 , 506 , 758
<code>\glswrglossdisableanchorcmds</code>		<code>\hypertarget</code>	132 , 448 , 700
.	202 , 206 , 236	<code>\hypertt</code>	237 , 506 , 758
<code>\glswrglossdisablelocationcmds</code>		<code>\hyperup</code>	237 , 506 , 758
.	197 , 200 , 201		
<code>\glswrglosslocationtarget</code>	202 , 236	I	
<code>\glswrglosslocationtextfmt</code>	202 , 236	<code>\IeC</code>	24 , 382 , 632
<code>\glswrite</code>		<code>\if</code>	124 , 206 , 442 , 487 , 571 , 693 , 739 , 824
.	178–183 , 191 , 193 , 198 , 469–	<code>\if@endfor</code>	269 , 591 , 844
.	473 , 479 , 483 , 484 , 571–575 ,	<code>\if@files</code>	32
.	721–725 , 731 , 735 , 736 , 825–828	<code>\if@glsw</code>	5 , 21 ,
<code>\glswritedefhook</code>			199 , 372 , 380 , 484 , 621 , 630 , 736
.	84 , 92 , 416 , 421 , 666 , 672	<code>\if@glsw</code>	4 ,
<code>\glswriteentry</code>	200 , 200 , 484 , 736		4 , 17 , 371 , 378 , 390 , 621 , 628 , 640
<code>\glswritefiles</code>		<code>\if@glsw</code>	
.	36 , 198 , 389 , 483 , 639 , 735		20 , 120 , 380 , 440 , 629 , 691
<code>\glsxindyfalse</code>	31 , 387 , 637	<code>\if@openright</code>	51 , 397 , 647
<code>\glsxindytrue</code>	31 , 32 , 387 , 637	<code>\if@org@KV@glsw</code>	118

<code>\ifbool</code>	36,	193, 220–222, 243, 245, 251,
	63, 379, 386, 389, 405, 432–	260, 345, 347, 362, 380, 388,
	434, 628, 636, 639, 655, 683–685	397, 404, 406, 407, 409, 432,
<code>\ifboolexpr</code>	44,	435, 436, 479, 480, 496, 497,
	69, 393, 408, 503, 643, 658, 755	508, 509, 514, 519, 523, 525,
<code>\ifcase</code>	5, 7, 8, 28, 34, 37, 39, 76,	527, 543, 546, 548, 559, 630,
	224, 315, 372–374, 385, 390,	638, 647, 654, 656, 657, 659,
	411, 499, 580, 616, 621, 623,	660, 683, 686, 687, 731, 732,
	624, 635, 640, 661, 750, 833, 870	748, 749, 760, 761, 766, 772,
<code>\ifcsdef</code>	28,	775, 777, 780, 796, 799, 801, 812
	45, 51, 79, 80, 87–92, 102, 200,	<code>\ifdefequal</code>
	215, 218–220, 245, 246, 269,	32, 65–68, 79, 80, 85,
	340, 385, 394, 397, 413, 414,	95, 99, 222, 341, 387, 388, 405–
	418–421, 438, 484, 492, 494,	407, 413, 414, 416, 423, 426,
	495, 504, 509, 540, 635, 644,	497, 541, 637, 638, 656, 657,
	647, 663, 664, 668–672, 689,	663, 664, 667, 674, 677, 749
	736, 744, 746, 747, 756, 761, 793	<code>\ifdefstrequal</code>
<code>\ifcsempy</code>	65,	91, 420, 671
	326, 327, 406, 587, 656, 840, 841	<code>\ifdefstring</code>
<code>\ifcsequal</code>	66, 406, 656	11, 44, 69, 188–190,
<code>\ifcsname</code>	91, 671	193, 217–219, 221, 223, 332,
<code>\ifcsstrequal</code>	92, 421, 672	375, 393, 408, 477–479, 494–
<code>\ifcsstring</code>	91, 420, 671	498, 535, 625, 643, 658, 728–
<code>\ifcsundef</code>	8, 15, 31, 41,	730, 732, 745, 746, 748, 749, 788
	44, 47, 49, 51, 62, 63, 71, 72,	<code>\ifdefvoid</code>
	75, 94, 96, 97, 106, 117, 123,	23, 24, 31, 43, 98,
	132, 133, 198, 210, 216, 224,	188, 189, 221, 222, 382, 425,
	228, 232–235, 238, 245, 269,	496, 497, 631, 632, 676, 748, 749
	341–344, 374, 377, 387, 392,	<code>\ifdim</code>
	393, 395–397, 404, 405, 408–	247, 287,
	410, 422, 424, 429, 439, 441,	319, 511, 600, 619, 763, 854, 872
	448, 449, 477, 478, 483, 489,	<code>\iffalse</code>
	491, 492, 498, 501, 503, 504,	98, 104, 425, 428, 676, 679
	506, 509, 541–543, 575, 591,	<code>\IfFileExists</code>
	623, 627, 637, 642, 644, 646,	10, 27, 28, 189–
	647, 655, 659, 661, 673, 675,	191, 213, 375, 385, 477, 478,
	680, 690, 692, 700, 729, 735,	491, 625, 634, 635, 729, 730, 742
	741, 743, 744, 750, 753, 755,	<code>\ifglossaryexists</code>
	756, 758, 761, 794–796, 829, 845	48, 60,
<code>\ifcsvoid</code>	67	62, 64, 187–190, 396, 403–405,
<code>\ifdef</code>	2, 40,	476–478, 646, 653–655, 728–730
	67, 77, 83, 84, 103, 118, 163,	<code>\ifgls@sanitize@description</code>
	164, 195, 196, 202, 239, 273,	25, 383, 633
	314, 327, 340, 407, 412, 415,	<code>\ifgls@sanitize@name</code>
	427, 439, 462, 464, 481, 482,	25, 383, 633
	506, 540, 588, 615, 616, 657,	<code>\ifgls@sanitize@symbol</code>
	662, 666, 678, 690, 714, 716,	26, 383, 633
	733, 734, 758, 793, 841, 847, 869	<code>\ifgls@xindy@glsnumbers</code>
<code>\ifdefempty</code>	20, 33, 50, 51, 62,	61, 404, 654
	66–68, 72, 112, 114, 190, 191,	<code>\ifglsacrdescription</code>
		30, 530, 784
		<code>\ifglsacrdua</code>
		30, 521,
		526, 529–531, 774, 779, 782, 784
		<code>\ifglsacrfootnote</code>
		29, 120, 440, 530, 531, 691, 784
		<code>\ifglsacronym</code>
		18, 378, 379, 628
		<code>\ifglsacrshortcuts</code>
		40, 257, 391, 518, 642, 770
		<code>\ifglsacrsmalldcaps</code>
		30, 521, 522, 524, 526,
		529, 773–775, 777, 779, 781, 782

<code>\ifglsmaller</code>	30, 521, 522, 525, 526, 773–775, 777, 779	<code>\ifglsnogroupskip</code>	274, 278, 279, 281, 284, 285, 289, 291, 292, 300, 302, 304, 308, 310, 312, 315, 317, 318, 321, 594, 596, 597, 599, 602, 603, 608–610, 612–614, 616–618, 620, 847, 849–853, 855–857, 862–864, 866–868, 870–872, 874
<code>\ifglautomake</code>	34, 35, 40, 193, 388, 389, 480, 638, 639, 731	<code>\ifglslonumberlist</code>	226, 499, 751
<code>\ifglsdscsuppressed</code>	66, 271, 406, 592, 656, 846	<code>\ifglslonopostdot</code>	11, 375, 625
<code>\ifglsentrycounter</code>	12, 13, 40, 227, 228, 376, 392, 500, 501, 625, 626, 642, 752, 753	<code>\ifglslnumberline</code>	52, 398, 648
<code>\ifglsentryexists</code>	63, 63, 64, 81–83, 93, 95, 404, 405, 415, 421, 423, 655, 665, 666, 672, 674	<code>\ifglslresetcurrcount</code>	106, 107
<code>\ifglssesclocations</code>	125, 200–202, 485, 737	<code>\ifglssanitizesort</code>	23, 26, 381, 384, 631, 633, 634
<code>\ifglsfieidcseq</code>	92, 421, 672	<code>\ifglssavenumberlist</code>	79, 192, 209, 413, 480, 489, 663, 731, 740
<code>\ifglsfieiddefeq</code>	91, 420, 671	<code>\ifglssavewrites</code>	36, 187, 200, 389, 476, 484, 639, 728, 736
<code>\ifglsfieidseq</code>	91, 420, 671	<code>\ifglssubentrycounter</code>	13, 40, 227, 228, 376, 392, 500, 501, 626, 642, 752, 753
<code>\ifglsfieidvoid</code>	66	<code>\ifglstoc</code>	52, 398, 648
<code>\ifglshaschildren</code>	65, 272, 405, 576, 593, 655, 829, 846	<code>\ifglstranslate</code>	27, 43, 385, 393, 634, 643
<code>\ifglshasdesc</code>	65, 271, 406, 593, 656, 846	<code>\ifglsucmark</code>	49, 50, 396, 646, 647
<code>\ifglshasfield</code>	67, 407, 657	<code>\ifglused</code>	63, 108, 112– 114, 120, 175, 200, 327–331, 345–347, 405, 431–436, 440, 468, 484, 520, 523, 525, 527, 543–548, 588–590, 655, 681, 683–687, 691, 719, 736, 772, 775, 778, 780, 796–801, 841–844
<code>\ifglshaslong</code>	66, 109–111, 164, 165, 246, 247, 251, 253, 359, 362, 365, 406, 431, 432, 463, 510, 511, 514, 516, 525, 556, 557, 559, 561, 656, 682, 683, 714, 762, 763, 766, 767, 778, 809, 810, 812, 814	<code>\ifglswrallowprimitivemods</code>	203, 204, 486, 738
<code>\ifglshasparent</code>	65, 216, 221, 319, 406, 492, 496, 618, 656, 744, 748, 872	<code>\ifglxindy</code>	13, 30, 46, 47, 52–61, 72, 99, 100, 125, 176, 177, 183, 188, 189, 191, 206, 207, 213, 387, 395, 398– 404, 409, 426, 443, 468, 469, 473, 476, 477, 487, 488, 491, 570, 571, 637, 645, 648–654, 659, 677, 694, 720, 721, 725, 728, 729, 738, 740, 743, 823–825
<code>\ifglshasprefix</code>	326, 327, 329, 330, 587–590, 840–843	<code>\ifignoredglossary</code>	73, 94, 98, 200, 210, 410, 422, 425, 484, 489, 660, 673, 676, 736, 741
<code>\ifglshasprefixfirst</code>	326, 327, 329, 330, 587–590, 841–843	<code>\ifin@</code>	39, 391, 641
<code>\ifglshasprefixfirstplural</code>	327, 328, 330, 331, 587, 588, 590, 591, 841–844	<code>\ifinlistcs</code>	219, 224, 495, 498, 747, 750
<code>\ifglshasprefixplural</code>	326, 328, 329, 331, 587–590, 840, 842–844	<code>\ifinner</code>	6, 372, 622
<code>\ifglshasshort</code>	66, 209, 406, 657		
<code>\ifglshassymbol</code>	66, 315, 316, 318, 320, 321, 406, 520, 523, 527, 616–620, 656, 772, 775, 780, 870, 871, 873		
<code>\ifglshyperfirst</code>	120, 440, 691		
<code>\ifglindexonlyfirst</code> . .	200, 484, 736		

<code>\ifKV@glslink@hyper</code>	<code>\immediate</code>
..... 121, 122, 440, 441, 691, 692	33, 82, 84, 108, 188–191, 198,
<code>\ifKV@glslink@local</code>	214, 388, 415, 416, 431, 476,
..... 118, 449–452, 701–703	478, 484, 491, 638, 665, 666,
<code>\ifmeasuring@</code> 101, 427, 678	681, 682, 728, 730, 735, 736, 743
<code>\ifmmode</code> 6, 372, 622	<code>\in@</code> 39, 391, 641
<code>\ifnum</code> 14,	<code>index (option)</code> 38
107, 108, 190, 221, 287, 316,	<code>index (style)</code> 314
318, 320, 321, 377, 388, 430,	<code>indexgroup (style)</code> 315
479, 496, 580–582, 600, 617–	<code>indexhypergroup (style)</code> 315
619, 626, 627, 639, 680, 681,	<code>\indexname</code> 39, 391, 641
730, 748, 834, 835, 854, 871, 873	<code>indexonlyfirst (option)</code> 29
<code>\ifstrempy</code> 7, 373, 576, 623, 829	<code>\indexspace</code> 273, 274, 294, 294, 295–
<code>\ifstrequal</code> 503, 755	299, 313, 315–319, 321, 322,
<code>\ifthenelse</code> 27, 41, 51,	593, 594, 604–608, 615–618,
123, 192, 269, 384, 392, 397,	620, 847, 858–861, 869–872, 874
441, 479, 503, 531, 591, 634,	<code>inline (style)</code> 270
642, 647, 692, 731, 755, 784, 844	<code>\input</code> 43, 111, 393, 432, 643, 683
<code>\IfTrackedLanguage</code> 184, 473, 725	<code>\inputencodingname</code> 31, 387, 637
<code>\IfTrackedLanguageFileExists</code> ..	<code>\InputIfFileExists</code> 83, 415, 666
..... 44, 323,	<code>\istfile</code> 197, 483, 735
324, 394, 569, 586, 644, 822, 839	<code>\istfilename</code> 46, 46,
<code>\iftrue</code> ... 98, 104, 425, 428, 676, 679	178, 182, 189, 190, 192, 395,
<code>\ifundef</code> 12, 13, 70,	469, 472, 477–479, 571, 574,
82, 94, 178, 182, 191, 376, 408,	645, 721, 724, 729–731, 825, 827
415, 422, 469, 472, 479, 625,	<code>\item</code> . 273–277, 294, 295, 314–316,
626, 658, 665, 673, 721, 724, 731	576, 577, 579, 580, 594, 595,
<code>\ifvmode</code> 175, 467, 719	604, 605, 616, 617, 829, 830,
<code>\ifvoid</code> 287, 601, 854	833, 847–849, 858, 859, 869, 870
<code>\ifx</code> 12, 14, 16, 19, 33, 38,	
39, 41, 53, 54, 56, 58, 60, 61,	
69, 95–97, 100, 102, 123, 126–	
132, 165, 178, 181, 183, 184,	
186, 198, 202, 205, 207, 208,	
211, 212, 233, 235, 236, 260,	
261, 332, 341, 376–380, 391,	
392, 398, 399, 401, 403, 404,	
422–426, 441, 443–448, 463,	
469, 471, 473–475, 483, 485–	
488, 490, 503–505, 519, 521,	
522, 524, 526–528, 530, 532,	
534, 540, 571–576, 580–582,	
625–627, 629, 638, 641, 642,	
648, 649, 651, 652, 654, 673–	
677, 692, 694–700, 714, 715,	
721, 723, 725–727, 735, 737–	
740, 742, 755–757, 771, 773,	
775, 777, 779–781, 783, 785,	
788, 794, 825, 827–829, 833–835	

J

<code>\jobname</code> 33, 46, 82, 83, 178,
182, 188–191, 213, 388, 395,
415, 469, 472, 476–478, 491,
571, 574, 638, 645, 665, 666,
721, 724, 728–730, 742, 825, 827

K

<code>kernelglossredefs (option)</code> 37
<code>\key@ifundefined</code> 85–87, 417, 667, 668
keys:
<code>access</code> 333
<code>counter</code> 75
<code>description</code> 73
<code>descriptionaccess</code> 334
<code>descriptionplural</code> 73
<code>descriptionpluralaccess</code> 334
<code>first</code> 74
<code>firstaccess</code> 333
<code>firstplural</code> 74

firstpluralaccess	333	\let	4, 5, 11, 14–16, 18, 24, 25, 27–
long	78		29, 31, 33–41, 44, 45, 55, 67–
longaccess	334		69, 79, 81, 92, 93, 95, 97, 98,
longplural	78		101–104, 106, 108, 111, 118–
longpluralaccess	334		122, 124, 125, 132–139, 155–
name	73		162, 165, 173, 182, 183, 187–
nonumberlist	76		190, 192, 194–196, 199, 201,
parent	76		202, 204, 208–213, 222, 224,
plural	74		226, 243, 256, 257, 269, 270,
pluralaccess	333		273, 274, 287, 294, 295, 314,
see	75		323, 325, 335, 336, 350–352,
short	78		371, 372, 375, 377–379, 382,
shortaccess	334		383, 385, 386, 388–393, 395,
shortplural	78		399, 400, 407, 408, 413, 414,
shortpluralaccess	334		421–425, 427–430, 432, 439–
sort	74		443, 448–452, 458–463, 466,
symbol	74		467, 472, 473, 476–482, 484–
symbolaccess	333		486, 488–490, 497, 498, 500,
symbolplural	75		502, 508, 518–530, 537, 550,
symbolpluralaccess	333		551, 564–569, 575, 586, 587,
text	74		591, 592, 600, 604, 605, 616,
textaccess	333		621, 622, 625–628, 632–636,
type	75		638–645, 650, 657, 658, 663–
user1	77		665, 672–676, 678–681, 683,
user1access	334		690–694, 700–704, 709–715,
user2	77		718, 719, 723, 725, 728–734,
user2access	334		736–738, 740–742, 749, 750,
user3	77		752, 754, 760, 770, 771, 773–
user3access	334		781, 783, 790, 791, 803–805,
user4	77		817–822, 828, 839, 840, 844,
user4access	335		845, 847, 854, 858, 859, 869
user5	77	\letcs	31, 33, 65–68,
user5access	335		84, 86, 87, 90, 96, 97, 163, 165,
user6	78		195, 196, 215–217, 221, 222,
user6access	335		229, 232, 270, 319, 405–407,
\KV@glslink@hyperfalse			415, 417, 420, 424, 462, 463,
118, 120, 121,			477, 478, 481, 482, 492, 493,
133, 439, 440, 449, 690, 691, 700			496, 497, 501, 503, 619, 638,
\KV@glslink@hypertrue			656, 657, 666, 668, 671, 674,
118, 133, 439, 449, 690, 700			675, 714, 729, 730, 733, 734,
			744, 745, 748, 749, 753, 755, 872
L		list (style)	274
\L	25, 383, 632	\listcsadd	219, 495, 747
\l	3, 25, 383, 632	\listcsgadd	224, 498, 750
\label	9, 101, 102, 225,	\listcsxadd	215, 492, 744
227, 374, 499, 500, 624, 751, 752		listdotted (style)	276
\language	31, 387, 637	\listead	219, 495, 747
languages (option)	29	listgroup (style)	274
\leaders	276, 577, 595, 830, 848	listhypergroup (style)	275
\leavevmode	92, 121, 421, 440, 672, 691	\listxadd	82, 665

<code>\loadglsentries</code> ..	111 , 111 , 432 , 683
<code>\long</code>	92 , 236 , 421 , 506 , 672 , 757
<code>long (key)</code>	78
<code>long (style)</code>	277
<code>long-booktabs (style)</code>	284
<code>long-sc-short (acrstyle)</code>	248
<code>long-sc-short-desc (acrstyle)</code>	249 , 360
<code>long-short (acrstyle)</code>	246 , 359
<code>long-short-desc (acrstyle)</code> ..	249 , 360
<code>long-sm-short (acrstyle)</code>	248
<code>long-sm-short-desc (acrstyle)</code>	250 , 361
<code>long-sp-short (acrstyle)</code>	247
<code>long-sp-short-desc (acrstyle)</code> ..	249
<code>long3col (style)</code>	279
<code>long3col-booktabs (style)</code>	284
<code>long3colborder (style)</code>	280
<code>long3colheader (style)</code>	280
<code>long3colheaderborder (style)</code>	280
<code>long4col (style)</code>	280
<code>long4col-booktabs (style)</code>	285
<code>long4colborder (style)</code>	281
<code>long4colheader (style)</code>	281
<code>long4colheaderborder (style)</code>	282
<code>longaccess (key)</code>	334
<code>longborder (style)</code>	278
<code>longheader (style)</code>	278
<code>longheaderborder (style)</code>	279
<code>\longnewglossaryentry</code> 92 , 93 , 421 , 672	
<code>longplural (key)</code>	78
<code>longpluralaccess (key)</code>	334
<code>\longprovideglossaryentry</code>	93 , 421 , 672
<code>longragged (style)</code>	288
<code>longragged-booktabs (style)</code>	286
<code>longragged3col (style)</code>	290
<code>longragged3col-booktabs (style)</code> .	286
<code>longragged3colborder (style)</code>	291
<code>longragged3colheader (style)</code>	291
<code>longragged3colheaderborder</code> <code>(style)</code>	291
<code>longraggedborder (style)</code>	289
<code>longraggedheader (style)</code>	289
<code>longraggedheaderborder (style)</code> ..	290
<code>\LT@end@pen</code>	287 , 600 , 854
<code>\LT@err</code>	287 , 600 , 854
<code>\LT@foot</code>	287 , 600 , 601 , 854
<code>\LT@head</code>	287 , 288 , 600 , 601 , 854
<code>\LT@lastfoot</code>	287 , 600 , 601 , 854
<code>\LT@output</code>	287 , 600 , 854
M	
<code>\makeatletter</code>	83 , 213 , 415 , 491 , 666 , 742
<code>\makebox</code>	276 , 319 – 321 , 577 , 581 , 582 , 595 , 619 , 620 , 830 , 835 , 848 , 872 , 873
<code>\makefirsttuc</code>	2
<code>\makeglossaries</code>	7 , 32 – 34 , 37 , 38 , 42 , 76 , 187 , 191 , 193 , 195 , 197 , 214 , 373 , 387 – 390 , 393 , 411 , 476 , 479 – 481 , 483 , 491 , 623 , 637 – 640 , 643 , 661 , 728 , 731 – 733 , 735 , 743
<code>\makeglossary</code>	36 , 36 , 37 , 38 , 389 , 390 , 639 , 640
<code>makeindex (option)</code>	31
<code>\makenoidxglossaries</code>	7 , 32 , 33 , 76 , 192 , 193 , 197 , 373 , 387 , 388 , 411 , 480 – 483 , 623 , 637 , 638 , 661 , 731 – 735
<code>\MakeTextUppercase</code>	3 , 371 , 621
<code>\MakeUppercase</code>	544 , 545 , 551 , 797 – 799 , 804 , 805
<code>\marginpar</code>	6 , 373 , 622
<code>\markboth</code>	49 , 50 , 396 , 646
<code>\mbox</code>	175 , 275 , 298 , 299 , 320 , 467 , 577 , 594 , 607 , 619 , 719 , 830 , 848 , 860 , 861 , 873
<code>mcolaltnotree (style)</code>	298
<code>mcolaltnotreegroup (style)</code>	298
<code>mcolaltnotreehypergroup (style)</code> ...	298
<code>mcolaltnotreespannav (style)</code>	299
<code>mcolindex (style)</code>	294
<code>mcolindexgroup (style)</code>	294
<code>mcolindexhypergroup (style)</code>	295
<code>mcolindexspannav (style)</code>	295
<code>mcoltree (style)</code>	295
<code>mcoltreegroup (style)</code>	296
<code>mcoltreehypergroup (style)</code>	296
<code>mcoltreenoname (style)</code>	297
<code>mcoltreenonamegroup (style)</code>	297
<code>mcoltreenonamehypergroup (style)</code>	297
<code>mcoltreenonamespannav (style)</code> ...	297
<code>mcoltreespannav (style)</code>	296
<code>\memUchead</code>	49 , 396 , 646
<code>\MessageBreak</code>	21 , 37 , 69 , 211 , 212 , 323 , 324 , 332 , 381 , 389 , 390 , 408 , 490 , 534 , 569 , 586 , 630 , 639 , 640 , 658 , 742 , 788 , 822 , 823 , 839

<code>mfirstuc</code> (option)	39		
<code>\mfirstucMakeUppercase</code>			
.	3 , 371 , 396 ,		
	419 , 433–437 , 452–462 , 509 ,		
	514–516 , 523 , 527 , 547–549 ,		
	552–555 , 559 , 560 , 562 , 590 ,		
	591 , 621 , 646 , 647 , 669 , 684–		
	688 , 704–712 , 714 , 760 , 761 ,		
	766–768 , 776 , 780 , 800–802 ,		
	805–808 , 812 , 813 , 815 , 843 , 844		
<code>\MFUaddmap</code>	3		
<code>\MFUblocker</code>	3		
<code>\MFUexcl</code>	3		
<code>\MFUsentencecase</code> 2 , 3 , 39 , 163 , 164 , 168			
<code>\midrule</code>	284 , 285 , 599 , 852 , 853		
<code>\month</code>	178 , 182 , 469 ,		
	472 , 571 , 574 , 721 , 724 , 825 , 827		
N			
<code>\n</code> . . .	182 , 472 , 574 , 575 , 724 , 827 , 828		
<code>name</code> (key)	73		
<code>\NeedsTeXFormat</code>	2 , 267 , 270 ,		
	273 , 277 , 283 , 288 , 294 , 299 ,		
	306 , 313 , 322–324 , 331 , 370 ,		
	371 , 534 , 569 , 570 , 575 , 586 ,		
	621 , 787 , 822 , 823 , 829 , 839 , 847		
<code>\new@glossaryentry</code>	81 ,		
	82 , 195 , 414 , 415 , 481 , 665 , 733		
<code>\new@ifnextchar</code>			
	70 , 87 , 88 , 109 , 110 , 134–138 ,		
	140–162 , 241 , 242 , 327–331 ,		
	408 , 418 , 419 , 431 , 449–462 ,		
	506 , 507 , 588–590 , 658 , 669 ,		
	682 , 701–713 , 758 , 759 , 841–843		
<code>\newacronym</code> 238 , 238 , 239 , 243 , 506 ,			
	508 , 519 , 521 , 522 , 524 , 526 ,		
	528 , 530 , 532 , 758 , 760 , 771 ,		
	773 , 775 , 777 , 779 , 781 , 783 , 785		
<code>\newacronymhook</code> 243 , 244 , 368 , 508 ,			
	519 , 521 , 523 , 524 , 526 , 529 ,		
	530 , 532 , 564 , 760 , 771 , 773 ,		
	775 , 777 , 779 , 782 , 783 , 785 , 817		
<code>\newacronymstyle</code>			
 245 , 246–251 , 253 , 255 ,		
	256 , 509–515 , 517 , 761–767 , 769		
<code>\newcommand</code>	2 , 3 , 6–31 , 33–		
	35 , 37–43 , 45–88 , 91–93 , 98–		
	106 , 108–112 , 114 , 116–122 ,		
	124 , 125 , 132–177 , 183 , 186–		
	191 , 193 , 197–202 , 204 , 206–		
	211 , 213–223 , 225–234 , 237–		
	247 , 256 , 258 , 260–270 , 272–		
	274 , 287 , 294 , 313 , 314 , 319 ,		
	325–331 , 333 , 337–341 , 358 ,		
	368–370 , 372–382 , 384–406 ,		
	408–421 , 425–432 , 435 , 436 ,		
	438–443 , 448–469 , 473 , 476–		
	480 , 482–489 , 491–504 , 506–		
	511 , 518–535 , 539–541 , 556 ,		
	564 , 568 , 569 , 575 , 587–594 ,		
	600 , 604 , 615 , 616 , 619 , 622–		
	632 , 634–636 , 638–643 , 645–		
	650 , 652–656 , 658–672 , 676–		
	683 , 686–694 , 700–720 , 725 ,		
	728–732 , 734–763 , 770–788 ,		
	792–794 , 809 , 817 , 821 , 822 ,		
	829 , 840–847 , 854 , 858 , 869 , 872		
<code>\newcount</code> . .	15 , 79 , 377 , 413 , 627 , 663		
<code>\newcounter</code>	12 , 13 , 376 , 625 , 626		
<code>\newenvironment</code>	228 , 501 , 753		
<code>\newglossary</code>			
 17 , 18 , 38 , 39 , 70 , 71 , 192 ,		
	378 , 379 , 390 , 391 , 408 , 409 ,		
	479 , 628 , 629 , 641 , 658 , 659 , 731		
<code>\newglossaryentry</code>	7 ,		
	39 , 78 , 81 , 81 , 106 , 244 , 373 ,		
	391 , 412 , 414 , 429 , 508 , 518 ,		
	520 , 521 , 523 , 525 , 527 , 529 ,		
	531 , 564–567 , 623 , 641 , 662–		
	665 , 680 , 760 , 770 , 772 , 774 ,		
	776 , 778 , 780 , 782 , 784 , 817–820		
<code>\newglossarystyle</code>	233 ,		
	270 , 274–286 , 288–312 , 314–		
	319 , 322 , 504 , 592 , 594–620 ,		
	756 , 845 , 847–853 , 855–872 , 874		
<code>\newif</code> 4 , 5 , 20 , 27 , 30 , 34 , 106 , 203 ,			
	371 , 372 , 380 , 385 , 387 , 388 ,		
	486 , 621 , 629 , 634 , 637 , 638 , 738		
<code>\newignoredglossary</code> . . .	72 , 409 , 659		
<code>\newlength</code>	132 , 276 , 277 ,		
	288 , 300 , 307 , 317 , 448 , 595 ,		
	601 , 608 , 612 , 617 , 700 , 848 ,		
	849 , 855 , 861 , 862 , 865 , 866 , 871		
<code>\newrobustcmd</code> . . .	6 , 7 , 65–67 , 81 ,		
	82 , 87–91 , 109 , 110 , 120 , 134–		
	164 , 168 , 174 , 175 , 208 , 209 ,		
	232 , 235 , 239–242 , 260 , 319 ,		
	327–330 , 341–344 , 373 , 405–		
	407 , 414 , 415 , 418 , 419 , 431 ,		
	439 , 449–468 , 488 , 489 , 506 ,		

507, 541–543, 587–590, 618, 622, 623, 655–657, 664, 665, 669, 682, 690, 701–719, 740, 758, 759, 794–796, 840–843, 872	noredefwarn (option)	<u>22</u>
\newterm	\normalbaselineskip . . .	287, 600, 854
\newtoks	nostyles (option)	<u>11</u>
243, 443, 476, 508, 694, 728, 760	nosuper (option)	<u>11</u>
\newwrite	notranslate (option)	<u>28</u>
33, 82, 178, 182, 187, 191, 388, 415, 469, 472, 476, 479, 638, 665, 721, 724, 728, 731	notree (option)	<u>11</u>
\nfss@text	nowarn (option)	<u>21</u>
\noalign	\ns@ACRfull	241, 507, 759
\nobreak	\ns@Acrfull	241, 507, 758
288, 577, 594, 601, 830, 848, 854	\ns@acrfull	241, 506, 758
\noexpand	\ns@ACRfullpl	242, 507, 759
41, 53, 54, 83, 97, 98, 102, 123, 125, 126, 131, 132, 173, 184–186, 188–190, 192, 201, 206, 209, 214, 216, 217, 236, 238, 244, 379, 392, 398, 399, 415, 424, 425, 438, 439, 441– 443, 448, 467, 473–478, 480, 485, 486, 489, 491, 493, 501, 502, 506, 508, 518, 520–525, 527–529, 531, 535, 564–568, 570, 629, 643, 649, 666, 675, 676, 689, 692–694, 699, 700, 718, 725–731, 737, 738, 741, 743, 745, 753, 754, 758, 760, 770, 772–774, 776, 778, 780– 782, 784, 785, 788, 817–821, 823	\ns@Acrfullpl	242, 507, 759
noglossaryindex (option)	\ns@acrfullpl	242, 507, 759
nogroupskip (option)	\ns@ACRlong	160, 461, 712
\nohyperpage	\ns@Acrlong	159, 460, 712
nohypertypes (option)	\ns@acrlong	158, 460, 711
\noindent	\ns@ACRlongpl	162, 462, 713
299, 317–319, 502, 605–607, 617, 618, 754, 859–861, 871, 872	\ns@Acrlongpl	161, 461, 713
\noist	\ns@acrlongpl	160, 461, 713
187, 476, 574, 575, 728, 827, 828	\ns@ACRshort	156, 458, 710
nolangwarn (option)	\ns@Acrshort	154, 155, 458, 709
nolist (option)	\ns@ACRshortpl	158, 459, 711
nolong (option)	\ns@Acrshortpl	157, 459, 711
nomain (option)	\ns@acrshortpl	156, 459, 710
nonumberlist (key)	\ns@newglossary	70, 70, 408, 658
nonumberlist (option)	\null	125–132, 184–186, 213, 442–448, 473– 476, 491, 693–700, 725–727, 743
\nopostdesc	\number	14, 83, 96, 102, 108, 201, 204, 205, 377, 415, 423, 430, 485, 486, 502, 535, 627, 666, 674, 681, 737, 738, 754, 788
45, 92, 212, 391, 394, 395, 421, 490, 576, 641, 645, 672, 742, 829	numberedsection (option)	<u>8</u>
nopostdot (option)	\numberline	52, 398, 648
<u>11</u>	numberline (option)	<u>8</u>
	numbers (option)	<u>38</u>
	\numexpr	108, 430, 681
	O	
	\O	25, 383, 632
	\o	25, 383, 633
	\OE	25, 383, 632
	\oe	25, 383, 632
	\oldacronym	237, 506, 758
	\openout	33, 82, 178, 182, 188, 388, 415, 469, 472, 476, 571, 574, 638, 665, 721, 724, 728, 825, 827
	options:	
	acronym	<u>18</u>

acronymlists	20	shortcuts	30
acronyms	18	smallcaps	30
automake	34	smaller	30
automakegloss	35	sort	13
automakeglosslite	35	style	9
compatible-2.07	36	subentrycounter	13
compatible-3.07	36	symbols	38
counter	20	toc	8
counterwithin	12	translate	28
debug	5	ucmark	12
description	30	writeglslabelnames	34
disablemakegloss	32	writeglslabels	34
dua	30	xindy	31
entrycounter	12	xindygloss	32
esclocations	10	xindynoglsnumbers	32
footnote	29	\OR	531, 784
hyperfirst	29	\or	5, 7–9, 28, 34, 35, 37–39, 225, 315, 372, 374, 386, 390, 499, 580, 616, 621– 624, 635, 640, 750, 751, 833, 870
index	38	order (option)	30
indexonlyfirst	29	\org@glossaryentrynumbers	212, 226, 490, 500, 742, 752
kernelglossredefs	37	\org@glossarytitle	211, 212, 490, 741, 742
languages	29	\org@glspostdescription	45, 395, 645
makeindex	31	\org@ifKV@glslink@hyper	121, 122, 440, 441, 691, 692
mfirsttuc	39	\outputpenalty	287, 600, 854
noglossaryindex	39		
nogroupskip	12		
nohypertypes	21		
nolangwarn	21		
nolist	11		
nolong	10		
nomain	18		
nonumberlist	10		
noposteddot	11		
noredefwarn	22		
nostyles	11		
nosuper	11		
notranslate	28		
notree	11		
nowarn	21		
numberedsection	8		
numberline	8		
numbers	38		
order	30		
restoremakegloss	32		
sanitize	27		
sanitizesort	26		
savenumberlist	10		
savewrites	36		
section	8		
seeautonumberlist	10		
seenoindex	7		

P

\p@	273, 294, 313, 314, 594, 604, 615, 616, 847, 858, 869
\p@glshyp@opt	119, 119, 439, 690
\PackageError	7, 16, 17, 25, 34, 42, 53, 60, 63, 64, 69, 75, 78, 79, 86–92, 94–96, 106, 117, 163, 186, 187, 192, 195, 197, 217–220, 224, 226, 233, 234, 239, 245, 246, 344, 370, 371, 373, 378, 389, 393, 399, 403, 405, 408, 410, 412, 413, 417– 423, 429, 439, 462, 476, 479, 481–483, 494, 495, 498, 499, 503, 504, 509, 510, 521, 522, 526, 529, 543, 576, 623, 628, 639, 643, 649, 653, 655, 658, 661–663, 668–674, 680, 690, 714, 728, 731, 733–735, 746,

747, 750, 751, 755, 756, 761, 762, 774, 775, 779, 782, 796, 829	379, 381, 390, 391, 480, 489, 499, 629, 631, 641, 732, 741, 751
\PackageInfo 5, 32, 33, 188, 199, 372, 388, 476, 484, 621, 622, 637, 638, 728, 736	\printindex 39, 391, 641
\PackageWarning 5, 21, 372, 380, 621, 622, 630	\printnoidxglossaries 194, 211, 481, 489, 733, 741
\PackageWarningNoLine 5, 21, 323, 324, 372, 380, 381, 569, 586, 621, 622, 630, 822, 839	\printnoidxglossary ... 194, 197, 211, 211, 217–219, 226, 480, 481, 483, 489, 494, 495, 499, 732, 733, 735, 741, 746, 747, 751
\pagegoal 287, 600, 854	\printnumbers 38, 391, 641
\pagelistname 42, 45, 280–282, 284, 285, 291, 293, 303–306, 310–313, 393, 394, 597–599, 603, 604, 610–612, 614, 615, 643, 644, 850, 851, 853, 856–858, 863–865, 867–869	\printsymbols 38, 390, 641
\par 45, 273, 275, 294, 296–299, 313, 314, 316–322, 395, 502, 503, 577, 580–582, 594, 604–608, 615– 620, 645, 754, 755, 830, 834, 835, 847, 848, 858–861, 869–874	\ProcessOptions 324, 331, 534, 586, 787, 839
parent (key) 76	\ProcessOptionsX 40, 391, 641
\parindent 294–299, 314, 316–318, 320–322, 580– 582, 604–608, 616–620, 833– 835, 858–861, 869–871, 873, 874	\protect 52, 116, 164, 246–248, 253, 254, 268, 349, 350, 359, 360, 365, 398, 438, 510, 511, 516, 520, 523, 525, 550, 556, 557, 561, 648, 688, 689, 762, 763, 768, 772, 776, 778, 803, 809, 810, 814
\parskip 294–298, 314, 316, 317, 604– 606, 616–618, 858–860, 869–871	\protected@csedef 89, 419, 670
\PassOptionsToPackage 324, 331, 534, 586, 787, 839	\protected@csxappto 269
\pdfstringdefPreHook 202	\protected@csxdef . 89, 269, 419, 670
\penalty 287, 600, 854	\protected@eappto 33, 72, 73, 99, 201, 638, 659, 660, 677, 737
\PGLS 330, 590, 843	\protected@edef 6, 9, 16, 19, 33, 54, 56, 59, 61, 62, 65, 69, 70, 72, 73, 89–95, 98– 100, 116, 117, 121, 123, 124, 164, 173–175, 190, 193, 194, 200, 203, 205, 206, 209, 215, 216, 225, 227, 233, 236, 244, 260, 268, 340, 341, 368, 373, 374, 399, 400, 402, 404, 423, 425, 426, 432–434, 438, 441, 442, 467, 484–486, 499, 501, 502, 504, 508, 527, 531, 535, 540, 541, 571, 572, 587, 591, 622, 624, 627, 629, 638, 649, 651, 653, 654, 656, 658–660, 670–674, 676, 677, 683–685, 689, 691–693, 718, 719, 732, 736–738, 741, 743, 744, 751– 754, 756, 760, 780, 784, 788, 793, 794, 817, 824, 825, 840, 844
\Pgl 328, 328, 589, 842	\protected@write 69, 71, 178–180, 191, 192, 194, 197, 200, 209, 268, 408, 469, 470, 479, 480, 483, 484, 489,
\pgls 327, 328, 588, 841	
\PGLSpl 330, 590, 843	
\PglSpl 329, 329, 589, 842	
\pglSpl 328, 329, 588, 841	
\phantomsection 51, 397, 647	
plural (key) 74	
pluralaccess (key) 333	
\printacronyms 18, 379, 628	
\PrintChanges 4, 371, 621	
\printglossaries 193, 211, 480, 489, 732, 741	
\printglossary 18, 22, 38, 39, 193, 210, 211, 226,	

572, 591, 658, 659, 721, 722,	374, 375, 377–379, 385, 386,
731, 732, 735, 736, 741, 825, 844	388, 390–392, 400, 408, 410,
<code>\protected@xdef</code> 14–	411, 414, 415, 423, 425, 427,
16, 19, 24, 80, 95, 98, 100, 101,	430, 439–448, 463, 471–476,
205, 213, 336, 337, 377–379,	479–482, 485–488, 490, 496,
382, 413, 426, 427, 486, 537–	498, 499, 503, 504, 532, 537,
539, 626, 627, 629, 632, 664,	550, 551, 564–567, 571, 573–
674, 676–678, 738, 742, 791, 792	575, 580–582, 591, 594, 600,
<code>\providecommand</code>	604, 607, 608, 615–621, 624,
. . . . 3, 18, 32, 42, 43, 50, 69,	625, 627, 628, 635, 638–642,
108, 133, 180, 191, 194, 214,	650, 658, 660, 661, 664, 666,
273, 294, 313, 327, 332, 379,	674, 676, 678, 680, 681, 690,
387, 393, 396, 408, 431, 449,	692–699, 714, 715, 723–725,
470, 479, 481, 483, 491, 501,	727, 728, 730–734, 737–740,
502, 535, 588, 593, 604, 615,	742, 748, 750, 755, 756, 785,
628, 637, 643, 647, 658, 681,	790, 791, 803, 804, 817–819,
701, 722, 731, 733, 735, 743,	821, 824, 826–828, 833–835,
753, 754, 788, 841, 847, 858, 869	844, 847, 854, 858, 861, 869–874
<code>\provideglossaryentry</code> . 81, 415, 665	<code>\renewacronymstyle</code> 246, 359–
<code>\ProvidesFile</code> 43, 393, 643	362, 364–367, 509, 556–559,
<code>\ProvidesGlossariesLang</code> 43, 393, 643	561–563, 761, 809–812, 814, 816
<code>\ProvidesPackage</code> . . . 2, 268, 270,	<code>\renewcommand</code> 3–5, 7–13, 16–18, 20–
273, 277, 283, 288, 294, 299,	22, 26, 28, 29, 32, 34–36, 38–
307, 313, 323, 324, 331, 370,	40, 44–47, 60, 73, 75, 77, 92,
371, 534, 569, 570, 575, 586,	102, 106–108, 173, 175–177,
591–593, 595, 599, 601, 604,	183, 184, 191–196, 213, 214,
608, 612, 615, 621, 787, 822,	224, 225, 239, 243, 244, 246–
823, 829, 839, 844, 845, 847,	256, 271, 272, 274–282, 284,
849, 852, 855, 858, 861, 865, 869	285, 287, 289–304, 307–312,
R	314–322, 332, 336, 345, 347,
<code>\r</code> 24, 382, 632	349, 350, 357–368, 371–376,
<code>\raggedright</code>	378–381, 384–391, 393–395,
. . . . 286–293, 307–313, 600–	403, 409, 411, 421, 429, 430,
604, 612–615, 853–858, 866–869	467–469, 473, 474, 479–482,
<code>\raisebox</code> 132, 448, 700	490, 491, 499, 508–519, 521,
<code>\ref</code> 227, 500, 752	522, 524–526, 528–530, 532,
<code>\refstepcounter</code> 101, 102, 227, 500, 752	535, 537, 543, 546, 548–550,
<code>\regex</code> 232	555–567, 570, 571, 575–585,
<code>\relax</code> 5, 8, 11,	592–610, 612–614, 616–626,
15, 16, 18, 28, 29, 33, 34, 37,	628–631, 633–637, 639, 640,
39, 40, 55, 69, 74, 76, 80, 83,	642, 644, 645, 653, 660–662,
95, 98, 103, 107, 108, 119, 120,	672, 680, 681, 718–720, 725,
123, 124, 126–131, 165, 180,	726, 731–734, 742, 743, 750,
182–184, 186, 187, 190, 192,	751, 760–775, 777, 779, 781–
194–196, 201, 205, 206, 208,	783, 785, 788, 790, 791, 796,
211, 213, 221, 224, 232, 233,	799, 801, 803, 808–820, 823,
261, 269, 273, 274, 287, 294,	824, 828–838, 845–864, 866–874
298, 299, 313, 315, 316, 318–	<code>\renewenvironment</code>
322, 335, 336, 350–352, 372, 228, 271, 274, 277–280,
	282, 283, 285–314, 316, 317,

320, 501, 592, 594–598, 600–	\setcounter 102, 227, 500, 752
619, 753, 845, 847, 849–871, 873	\SetCustomDisplayStyle
\renewglossarystyle . . . 234, 504, 756 261, 531, 532, 784, 785
\RequireGlossariesLang	\SetCustomStyle 261, 532, 785
. 43, 44, 323, 324, 393,	\SetDefaultAcronymDisplayStyle
394, 569, 586, 643, 644, 822, 839 258, 518, 519, 770, 771
\RequirePackage 2–	\SetDefaultAcronymStyle
4, 10, 11, 27, 28, 40, 43, 261, 258, 519, 530, 771, 783
277, 283, 288, 294, 299, 307,	\SetDescriptionAcronymDisplayStyle
323, 324, 332, 371, 375, 385, 259, 523, 524, 775, 777
391, 393, 532, 534, 535, 569,	\SetDescriptionAcronymStyle . . .
586, 595, 599, 601, 604, 608, 259, 524, 530, 777, 784
612, 621, 624, 625, 634, 635,	\SetDescriptionDUAAcronymDisplayStyle
641, 643, 785, 787, 788, 822, 259, 521–523, 774, 775
839, 849, 852, 855, 858, 861, 865	\SetDescriptionDUAAcronymStyle
\restorecounters@ 123, 441, 692 259, 522, 530, 775, 784
restoremakegloss (option) 32	\SetDescriptionFootnoteAcronymDisplayStyle
\romannumeral 201, 204, 258, 519, 521, 772–774
205, 319, 321, 485, 486, 582,	\SetDescriptionFootnoteAcronymStyle
619, 620, 737, 738, 835, 872, 873 259, 521, 530, 773, 784
 S	\SetDUADisplayStyle
\s@glshyp@opt . . . 119, 119, 439, 690 260, 529, 530, 782, 783
\s@GlsSetXdyFirstLetterAfterDigits	\SetDUASyle . . 260, 530, 531, 783, 784
. 176, 468, 720	\setentrycounter 54,
\s@GlsSetXdyNumberGroupOrder . .	180, 223, 233, 399, 470, 498,
. 177, 468, 469, 720	503, 570, 649, 722, 750, 755, 823
\s@ifglossaryexists 62,	\SetFootnoteAcronymDisplayStyle
62, 210, 404, 489, 654, 655, 741 259, 525, 526, 777, 779
\s@newglossary 70, 70, 408, 658	\SetFootnoteAcronymStyle
sanitize (option) 27 259, 526, 531, 779, 784
sanitizesort (option) 26	\SetGenericNewAcronym
\savecounters@ 123, 441, 692 243, 245, 508, 509, 760, 761
savenumberlist (option) 10	\setglossarypreamble . . 48, 396, 646
savewrites (option) 36	\setglossarysection . . . 50, 397, 647
sc-short-long (acrstyle) 248	\setglossarystyle 212, 224,
sc-short-long-desc (acrstyle) 250, 361	233, 261, 275–286, 289–299,
section (option) 8	301–306, 308–310, 312, 315,
see (key) 75	317, 318, 322, 490, 503, 504,
seeautonumberlist (option) 10	532, 594–600, 602–618, 620,
\seename . . 43, 208, 393, 488, 643, 740	742, 755, 756, 785, 847–872, 874
seenoindex (option) 7	\setglossentrycompatibility . . .
\setabbreviationstyle 260, 783	230, 498, 502, 504, 750, 754, 756
\SetAcronymLists 20, 380, 629	\setkeys 27,
\SetAcronymStyle 260,	31, 40, 50, 94, 122, 174, 175,
386, 387, 530, 531, 636, 783, 784	212, 243, 384, 387, 392, 397,
\setacronymstyle	422, 440, 467, 468, 490, 508,
. 239, 245, 260, 509, 761, 783	519, 521, 522, 524, 526, 528,
\setbool 26, 384, 634	530, 532, 634, 637, 642, 647,
\setbox 287, 600, 601, 854	673, 691, 719, 742, 760, 771,
	773, 775, 777, 779, 781, 783, 785

\setlength		\showgloplural	262, 532, 785
	. 276, 277, 288, 294–298, 300,		\showglopluralaccess	..	369, 568, 821
	307, 314, 316, 317, 321, 582,		\showgloshort	265, 533, 786
	595, 601, 604–606, 608, 612,		\showgloshortaccess	...	370, 569, 822
	616–620, 835, 848, 849, 855,		\showgloshortpluralaccess	
	858–862, 865, 866, 869–871, 873			370, 569, 822
\SetSmallAcronymDisplayStyle	..		\showglosort	264, 533, 786
 260, 527–529, 780–782		\showglossaries	266, 534, 787
\SetSmallAcronymStyle		\showglossarycounter	..	267, 534, 787
 260, 528, 531, 781, 784		\showglossaryentries	..	267, 534, 787
\setstylefile	45, 395, 645	\showglossaryin	266, 534, 787
\settodepth	103	\showglossaryout	266, 534, 787
\settoheight	102, 448, 700	\showglossarytitle	266, 534, 787
\settowidth	... 103, 247, 511, 581,		\showglosymbol	264, 533, 786
	582, 619, 620, 763, 835, 872, 873		\showglosymbolaccess	..	369, 568, 822
\setupglossaries	40, 391, 642	\showglosymbolplural	..	265, 533, 786
\sfcode	11, 375, 625	\showglosymbolpluralaccess	
short (key)	78		369, 568, 822
short-long (acrstyle)	247, 359	\showglotext	262, 532, 785
short-long-desc (acrstyle)	..	250, 361	\showglotextaccess	369, 568, 821
shortaccess (key)	334	\showglotype	262, 532, 786
shortcuts (option)	30	\showglouseri	263, 533, 786
shortplural (key)	78	\showglouserii	263, 533, 786
shortpluralaccess (key)	334	\showglouseriii	263, 533, 786
\show 261–267, 369, 370, 532–		\showglouseriv	263, 533, 786
	534, 568, 569, 785–787, 821, 822		\showglouserv	263, 533, 786
\showacronymlists	266, 534, 787	\showglouservi	264, 533, 786
\showglocounter	263, 532, 786	sm-short-long (acrstyle)	249
\showglodesc	264, 533, 786	sm-short-long-desc (acrstyle)		251, 362
\showglodescaccess	369, 569, 822	smallcaps (option)	30
\showglodescplural	264, 533, 786	smaller (option)	30
\showglodescpluralaccess		\SmallNewAcronymDef	260,
 369, 569, 822			368, 527, 529, 567, 780, 782, 820	
\showglofield	266, 534, 787	sort (key)	74
\showglofirst	262, 532, 786	sort (option)	13
\showglofirstaccess	..	369, 568, 821	\space	7, 32–
\showglofirstpl	262, 532, 786		34, 37, 42, 53, 54, 58, 61, 69,	
\showglofirstpluralaccess			76, 78, 106, 109–111, 116, 172,	
 369, 568, 822			178–182, 186, 188–194, 197,	
\showgloflag	265, 533, 787		202, 209, 212, 214, 228, 240,	
\showgloindex	265, 533, 787		246–256, 260, 270, 272, 274,	
\showglolevel	262, 532, 785		275, 278, 289, 300, 307, 308,	
\showgloloclist	265, 534, 787		314–316, 318, 320, 321, 349,	
\showglolong	265, 533, 787		350, 359–368, 373, 387–390,	
\showglolongaccess	370, 569, 822		393, 399, 402, 404, 411, 412,	
\showglolongpluralaccess			429, 431, 432, 438, 439, 466,	
 370, 569, 822			469–472, 476–483, 488, 490,	
\showglongname	264, 533, 786		491, 500, 501, 504, 507, 510–	
\showglongnameaccess	369, 568, 821		518, 523, 527, 549, 550, 556–	
\showgloparent	261, 532, 785		564, 570–574, 576–584, 592–	

596, 601, 608, 612, 616–620, 623, 637–640, 643, 649, 652, 654, 661–663, 680, 682, 688– 690, 718, 721–724, 728–735, 740, 742, 743, 752, 756, 758, 762–770, 776, 780, 803, 809– 817, 823–827, 830, 832–838, 845–849, 855, 862, 866, 869–873		altlong4colheader	282
\spacefactor	11, 375, 625	altlong4colheaderborder	283
\SS	25, 383, 633	altlongragged4col	292
\ss	25, 383, 633	altlongragged4col-booktabs .	286
\str	233	altlongragged4colborder	293
\string 7, 16, 17, 22, 25, 32–34, 37, 42, 52–54, 56–59, 61, 69, 71, 76, 78, 84, 87, 88, 99, 100, 106, 108–111, 124, 125, 127, 128, 131, 172, 176–183, 186, 187, 191–195, 197, 202, 206–208, 212, 214, 217–219, 226, 239, 260, 268, 370, 371, 373, 378, 381, 387–390, 393, 398–404, 408, 411, 412, 415, 416, 418, 426, 429, 431, 432, 438, 439, 442, 444, 445, 447, 466, 468– 473, 475, 476, 479–483, 487, 488, 490, 491, 494, 495, 499, 502–504, 570–575, 591, 623, 628, 631, 637–640, 643, 648, 649, 651–654, 658, 659, 661– 663, 666–669, 677, 680–682, 689, 690, 693, 695–697, 699, 718–725, 727, 728, 731–735, 739, 740, 742, 743, 746, 747, 751, 754, 756, 783, 823–828, 844		altlongragged4colheader	292
\strut	274–276, 278, 279, 281, 289, 290, 292, 300, 302, 304, 308, 309, 311, 318, 503, 576–579, 581, 583– 585, 594–597, 601–603, 608– 610, 612–614, 618, 755, 830– 832, 834, 836–838, 847–851, 855–857, 862–864, 866–868, 872	altlongragged4colheaderborder	293
style (option)	9	altsuper4col	305
styles:		altsuper4colborder	306
altlist	275	altsuper4colheader	305
altlistgroup	275	altsuper4colheaderborder . . .	306
altlisthypergroup	276	altsuperragged4col	311
altlong4col	282	altsuperragged4colborder . . .	312
altlong4col-booktabs	285	altsuperragged4colheader . . .	312
altlong4colborder	283	altsuperragged4colheaderborder	312
		alttree	319
		alttreegroup	322
		alttreehypergroup	322
		index	314
		indexgroup	315
		indexhypergroup	315
		inline	270
		list	274
		listdotted	276
		listgroup	274
		listhypergroup	275
		long	277
		long-booktabs	284
		long3col	279
		long3col-booktabs	284
		long3colborder	280
		long3colheader	280
		long3colheaderborder	280
		long4col	280
		long4col-booktabs	285
		long4colborder	281
		long4colheader	281
		long4colheaderborder	282
		longborder	278
		longheader	278
		longheaderborder	279
		longragged	288
		longragged-booktabs	286
		longragged3col	290
		longragged3col-booktabs	286
		longragged3colborder	291
		longragged3colheader	291
		longragged3colheaderborder .	291

longraggedborder	289	274–276, 278, 279, 281, 289,
longraggedheader	289	290, 292, 300, 302, 304, 308,
longraggedheaderborder	290	309, 311, 315, 316, 318, 320,
mcolalttree	298	426, 490, 496, 497, 502, 503,
mcolalttreegroup	298	593–597, 601–603, 608–610,
mcolalttreehypergroup	298	612–614, 616–619, 677, 742,
mcolalttreespannav	299	748, 754, 846–851, 855–857,
mcolindex	294	862–864, 866–868, 870, 871, 873
mcolindexgroup	294	\subitem 294, 314, 315, 580,
mcolindexhypergroup	295	604, 615, 616, 833, 858, 869, 870
mcolindexspannav	295	sublistdotted (style) 277
mcoltree	295	\subsubitem 294, 314, 315,
mcoltreegroup	296	580, 604, 616, 833, 858, 869, 870
mcoltreehypergroup	296	super (style) 300
mcoltreenoname	297	super3col (style) 302
mcoltreenonamegroup	297	super3colborder (style) 302
mcoltreenonamehypergroup	297	super3colheader (style) 303
mcoltreenonamespannav	297	super3colheaderborder (style) . . . 303
mcoltreespannav	296	super4col (style) 303
sublistdotted	277	super4colborder (style) 304
super	300	super4colheader (style) 304
super3col	302	super4colheaderborder (style) . . . 305
super3colborder	302	superborder (style) 301
super3colheader	303	superheader (style) 301
super3colheaderborder	303	superheaderborder (style) 301
super4col	303	superragged (style) 307
super4colborder	304	superragged3col (style) 309
super4colheader	304	superragged3colborder (style) . . . 310
super4colheaderborder	305	superragged3colheader (style) . . . 310
superborder	301	superraggedborder (style) 308
superheader	301	superraggedheader (style) 308
superheaderborder	301	superraggedheaderborder (style) . . 309
superragged	307	superraggedright3colheaderborder
superragged3col	309	(style) 310
superragged3colborder	310	symbol (key) 74
superragged3colheader	310	symbolaccess (key) 333
superraggedborder	308	\symbolname 42, 45, 281,
superraggedheader	308	282, 285, 293, 304–306, 312,
superraggedheaderborder	309	313, 393, 394, 598, 599, 604,
superraggedright3colheaderborder	310	610–612, 615, 643, 644, 851,
		853, 857, 858, 864, 865, 868, 869
tree	316	symbolplural (key) 75
treegroup	317	symbolpluralaccess (key) 333
treehypergroup	317	symbols (option) 38
treenoname	317	
treenonamegroup	318	
treenonamehypergroup	318	
subentrycounter (option)	13	
\subglossentry		
	99, 213, 221, 231, 272,	

T

\t	24, 382, 632
\tablehead	300–313, 608–615, 862–869
\tabletail	300–313, 608–615, 862–869

<code>\tabularnewline</code>	278–282, 284, 285, 289–293, 300–313, 578, 579, 583, 584, 595–599, 601–604, 608–615, 832, 836, 837, 849–853, 855–858, 862–869
<code>\texorpdfstring</code> . .	164, 202, 464, 716
<code>\text</code>	2, 3
<code>text (key)</code>	74
<code>textaccess (key)</code>	333
<code>\textbar</code>	270, 592, 845
<code>\textbf</code> . .	237, 313, 502, 506, 579– 582, 615, 754, 758, 833–835, 869
<code>\textit</code>	237, 506, 758
<code>\textmd</code>	237, 506, 758
<code>\textrm</code>	237, 506, 757
<code>\textsc</code>	237, 248, 249, 255, 366, 506, 512, 517, 521, 524, 526, 529, 563, 758, 763, 764, 769, 774, 777, 779, 782, 816
<code>\textsf</code>	237, 506, 758
<code>\textsl</code>	237, 506, 758
<code>\textsmaller</code>	248, 249, 255, 367, 512, 517, 521, 525, 526, 529, 563, 764, 769, 774, 777, 779, 782, 816
<code>\texttt</code>	237, 506, 758
<code>\textulc</code>	239, 506, 758
<code>\textup</code>	237, 239, 506, 758
<code>\TH</code>	25, 383, 633
<code>\th</code>	25, 383, 633
<code>\the</code>	41, 44, 54, 59, 61, 70, 126–132, 178, 182, 184, 186, 198, 200, 204, 205, 209, 221, 236, 244, 246– 248, 252, 253, 359, 363, 365, 368, 392, 394, 399, 402, 404, 408, 443–448, 469, 472, 474, 475, 483, 484, 486, 489, 496, 501, 502, 505, 508–511, 515, 516, 518–529, 531, 535, 556, 557, 560, 561, 564–568, 570, 571, 574, 643, 644, 649, 653, 654, 658, 694–700, 721, 724– 727, 735, 736, 738, 741, 748, 753, 754, 757, 760–763, 766, 768, 770–774, 776, 778, 780– 782, 784, 785, 788, 809, 810, 813, 814, 817–821, 823, 825, 827
<code>\the@numberlist</code> . .	173, 467, 718, 719
<code>\theglossary</code>	4, 371, 621
<code>theglossary (env.)</code>	228
<code>\theglossaryentry</code>	12, 228, 376, 500, 625, 752
<code>\theglossarysubentry</code>	13, 228, 376, 501, 626, 752
<code>\theglentrycounter</code>	123, 202, 205, 441, 485, 486, 571, 692, 737, 738, 824
<code>\theH</code>	207, 487, 739
<code>\theHglossaryentry</code>	12, 376, 625
<code>\theHglossarysubentry</code> .	13, 376, 626
<code>\theHglentrycounter</code>	123, 202, 205, 441, 442, 485, 486, 692, 737, 738
<code>\thesection</code>	41, 392, 642
<code>\this@dialect</code>	44, 323, 324, 394, 569, 586, 644, 822, 839
<code>\tiny</code>	6, 373, 622
<code>\tl</code>	3
<code>toc (option)</code>	8
<code>\toks@</code> .	41, 44, 54, 59, 61, 70, 126– 132, 184, 186, 209, 236, 392– 394, 399, 402, 404, 408, 443– 448, 473–475, 489, 501, 502, 505, 535, 570, 643, 644, 649, 653, 654, 658, 694–700, 725– 727, 741, 753, 754, 757, 788, 823
<code>\toprule</code>	284, 285, 599, 852, 853
<code>\TrackIfKnownLanguage</code>	43
<code>\trans@languages</code>	44, 393, 643
<code>\translate</code>	44, 45, 394, 644, 645
<code>translate (option)</code>	28
<code>\translatelet</code>	17, 18, 38, 378, 379, 390, 391, 628, 629, 641
<code>tree (style)</code>	316
<code>treegroup (style)</code>	317
<code>treehypergroup (style)</code>	317
<code>treenoname (style)</code>	317
<code>treenonamegroup (style)</code>	318
<code>treenonamehypergroup (style)</code>	318
<code>\triangleright</code>	6, 373, 622
<code>\ttfamily</code>	6, 373, 622
<code>\TX@trial</code>	103, 427, 678
<code>type (key)</code>	75
<code>\typeout</code>	21, 380, 630
U	
<code>\u</code>	24, 382, 632
<code>\uccode</code>	221, 496, 748
<code>ucmark (option)</code>	12
<code>\undef</code>	77, 83, 210, 411, 415, 489, 662, 666, 741

\unexpanded	39
\unskip	92, 276, 421, 577, 595, 672, 830, 848
\unvbox	287, 600, 601, 854
\usedictionary	44, 393, 644
\usepackage	25, 217–219, 239, 370, 371, 494, 495, 746, 747
user1 (key)	<u>77</u>
user1access (key)	<u>334</u>
user2 (key)	<u>77</u>
user2access (key)	<u>334</u>
user3 (key)	<u>77</u>
user3access (key)	<u>334</u>
user4 (key)	<u>77</u>
user4access (key)	<u>335</u>
user5 (key)	<u>77</u>
user5access (key)	<u>335</u>
user6 (key)	<u>78</u>
user6access (key)	<u>335</u>
\UTFviii@two@octets ...	25, 383, 633
\UTFviii@two@octets@combine	25, 383, 633
V	
\v	24, 383, 632
\value	102
\vbox	287, 600, 601, 854
\vsize	287, 288, 601, 854
\vskip ...	273, 287, 294, 313, 594, 600, 604, 615, 847, 854, 858, 869
\vss	287, 600, 601, 854
W	
\warn@nomakeglossaries ...	<u>190</u> , 192, 193, 195, 479–481, 730–733
\warn@noprintglossary	192–194, 210, 213, 479– 481, 489, 490, 731–733, 741, 742
X	
\write	33, 84, 108, 178–183, 188–191, 194, 198, 214, 388, 415, 416, 431, 469–473, 477, 478, 481, 484, 491, 571–575, 638, 666, 681, 682, 721–725, 729, 730, 733, 735, 743, 825–828
writelabelnames (option)	<u>34</u>
writelabels (option)	<u>34</u>
\writeint	177, 187, 469, 472, 473, 476, 571, 574, 575, 721, 723, 725, 728, 825, 827, 828
Y	
\x	505
\xatlevel@	123, 441, 692
\xcapitalisewords	464
\xdef	96, 423, 425, 490, 591, 674, 844, 845
\xglaccsupp ..	341, <u>341</u> , 540, 541, 794
\xglfieldaccsupp	<u>340</u> , 540, 793
\xifinlist	82, 665
\xifinlistcs	215, 216, 219, 492, 495, 743, 744, 747
xindy (option)	<u>31</u>
xindygloss (option)	<u>32</u>
xindynoglsnumbers (option)	<u>32</u>
\xmakefirstuc	432–434, 438, 462, 463, 587, 683–685, 689, 714, 715, 840
\xspace	238, 506, 758
Z	
\year	178, 182, 469, 472, 571, 574, 721, 724, 825, 827
Z	
\Z	232
\z@	287, 600, 601, 854