

Documented Code For glossaries v4.53

Nicola L.C. Talbot

Dickimaw Books

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

2023-09-29

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 (<code>glossary-hypernav.sty</code>) . . .	269
3.2	In-line Style (<code>glossary-inline.sty</code>)	273
3.3	List Style (<code>glossary-list.sty</code>)	276
3.4	Glossary Styles using <code>longtable</code> (<code>glossary-long.sty</code>)	280
3.5	Glossary Styles using <code>longtable</code> and <code>booktabs</code> (<code>glossary-longbooktabs.sty</code>)	286
3.6	Glossary Styles using <code>longtable</code> (<code>glossary-longragged.sty</code>)	291
3.7	Glossary Styles using <code>multicol</code> (<code>glossary-mcols.sty</code>)	297
3.8	Glossary Styles using <code>supertabular</code> environment (<code>glossary-super.sty</code>)	302
3.9	Glossary Styles using <code>supertabular</code> environment (<code>glossary-superragged.sty</code>)	309
3.10	Tree Styles (<code>glossary-tree.sty</code>)	316
4	Multi-Lingual Support	325
4.1	Babel Captions (<code>glossaries-babel.sty</code>)	325
4.2	Polyglossia Captions (<code>glossaries-polyglossia.sty</code>)	326
5	Prefix Support (<code>glossaries-prefix</code> Code)	327
6	Accessibility Support (<code>glossaries-accsupp</code> Code)	334
6.1	Defining Replacement Text	336
6.2	Accessing Replacement Text	340
6.3	Displaying the Glossary	360
6.4	Acronyms	362
6.5	Debugging Commands	372
7	Backwards Compatibility	373
7.1	<code>glossaries-compatible-207</code>	373
7.2	<code>glossaries-compatible-307</code>	373
8	Rollback	374
8.1	Rollback v4.46 (<code>glossaries-2020-03-19.sty</code>)	374
8.2	Rollback v4.46 (<code>glossaries-accsupp-2020-03-19.sty</code>)	537
8.3	Rollback v4.46 (<code>glossaries-babel-2020-03-19.sty</code>)	572
8.4	Rollback v4.46 (<code>glossaries-compatible-207-2020-03-19.sty</code>)	573
8.5	Rollback v4.46 (<code>glossaries-compatible-307-2020-03-19.sty</code>)	578
8.6	Rollback v4.46 (<code>glossaries-polyglossia-2020-03-19.sty</code>)	589
8.7	Rollback v4.46 (<code>glossaries-prefix-2020-03-19.sty</code>)	589
8.8	Rollback v4.46 (<code>glossary-hypernav-2020-03-19.sty</code>)	594
8.9	Rollback v4.46 (<code>glossary-inline-2020-03-19.sty</code>)	595
8.10	Rollback v4.46 (<code>glossary-list-2020-03-19.sty</code>)	596
8.11	Rollback v4.46 (<code>glossary-list-2020-03-19.sty</code>)	598
8.12	Rollback v4.46 (<code>glossary-longbooktabs-2020-03-19.sty</code>)	601
8.13	Rollback v4.46 (<code>glossary-longragged-2020-03-19.sty</code>)	604
8.14	Rollback v4.46 (<code>glossary-mcols-2020-03-19.sty</code>)	607
8.15	Rollback v4.46 (<code>glossary-super-2020-03-19.sty</code>)	611
8.16	Rollback v4.46 (<code>glossary-superragged-2020-03-19.sty</code>)	615

8.17	Rollback v4.46 (glossary-tree-2020-03-19.sty)	618
8.18	Rollback v4.49 (glossaries-2021-11-01.sty)	624
8.19	Rollback v4.49 (glossaries-accsupp-2021-11-01.sty)	790
8.20	Rollback v4.49 (glossaries-babel-2021-11-01.sty)	825
8.21	Rollback v4.49 (glossaries-compatible-207-2021-11-01.sty)	826
8.22	Rollback v4.49 (glossaries-compatible-307-2021-11-01.sty)	832
8.23	Rollback v4.49 (glossaries-polyglossia-2021-11-01.sty)	842
8.24	Rollback v4.49 (glossaries-prefix-2021-11-01.sty)	842
8.25	Rollback v4.49 (glossary-hypernav-2021-11-01.sty)	847
8.26	Rollback v4.49 (glossary-inline-2021-11-01.sty)	848
8.27	Rollback v4.49 (glossary-list-2021-11-01.sty)	850
8.28	Rollback v4.49 (glossary-long-2021-11-01.sty)	852
8.29	Rollback v4.49 (glossary-longbooktabs-2021-11-01.sty)	855
8.30	Rollback v4.49 (glossary-longragged-2021-11-01.sty)	858
8.31	Rollback v4.49 (glossary-mcols-2021-11-01.sty)	861
8.32	Rollback v4.49 (glossary-super-2021-11-01.sty)	864
8.33	Rollback v4.49 (glossary-superragged-2021-11-01.sty)	868
8.34	Rollback v4.49 (glossary-tree-2021-11-01.sty)	872
8.35	Rollback v4.52 (glossaries-2022-11-03.sty)	877
8.36	Rollback v4.52 (glossary-hypernav-2022-11-03.sty)	1031

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` v4.53: L^AT_EX₂ ϵ Package to Assist Generating Glossaries”.

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

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

INSTALL Installation instructions.

CHANGES Change log.

README.md Package summary.

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

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

2 Main Package Code (glossaries.sty)

2.1 Package Definition

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossaries-2020-03-19.sty}
```

```
\DeclareRelease{v4.49}{2021-11-01}{glossaries-2021-11-01.sty}
```

```
\DeclareRelease{v4.52}{2022-11-03}{glossaries-2022-11-03.sty}
```

```
\DeclareCurrentRelease{v4.53}{2023-09-29}
```

Declare package:

```
\ProvidesPackage{glossaries}[2023/09/29 v4.53 (NLCT)]
```

Required packages:

```
\RequirePackage{ifthen}
```

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

```
\RequirePackage{mfirstuc}
```

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

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

```
\ExplSyntaxOn
```

`\glslowercase` Provided to make it easier for user to use \LaTeX3 lowercase command without switching to `exp3` syntax. For example, to redefine `\acronymfont` to automatically convert to lowercase.

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

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

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

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

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

Automatically identify exclusions, blockers and mappings.

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

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

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

}
{
```

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

```
\RequirePackage{textcase}
```

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

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

Provide support for exclusions with `\MFUsentencecase`.

```
\glsmfuexcl
\providecommand{\glsmfuexcl}[1]{
\tl_if_in:NnF \l_text_case_exclude_arg_tl {#1}
{
\tl_put_right:Nn \l_text_case_exclude_arg_tl {#1}
}
}
```

`\mfirstucMakeUppercase` We have an old version of mfirstuc that is still using this command in the definition of `\makefirstuc`, in which case, continue to use `\MakeTextUppercase` as in previous versions, in case there is also an old version of textcase installed. However, note that the all-caps commands like `\GLS` now use `\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@gldebug
\define@choicekey{glossaries.sty}{debug}{\gldebug@val\gldebug@nr}%
{true,false,showtargets,showaccsupp}[true]{%
\ifcase\gldebug@nr\relax
% debug=true
\@gldebugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\let\@glshowtarget\@gobble
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\or
% debug=false
\@gldebugfalse
\let\@glshowtarget\@gobble
\let\@glshowaccsupp\@gobblethree
\PackageInfo{glossaries}{debug mode OFF}%
\or
% debug=showtargets
\@gldebugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glshowtarget}{\@glshowtarget}%
\or
% debug=showaccsupp
\@gldebugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glshowaccsupp}{\glshowaccsupp}%
```

```

    \fi
  }

```

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

```

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

```

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

```

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

```

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

```

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

```

`\glsshowtargetouter`

```

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

```

`\glsshowtargetsymbol`

```

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

```

`\glsshowtargetfont`

```

\newcommand*\glsshowtargetfont{\ttfamily\footnotesize}

```

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

```

\newcommand*\@glsshowtarget[1]{%

```

`\glsdoshowtarget` Make it easier to customise.

```

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

```

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

```

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

```


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

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

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

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

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

`\@gls@see@noindex`

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

`seenoindex`

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

```

\fi
}

```

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

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

```

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

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

```

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

```

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

```

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

```

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

```

Determine whether or not to use numbered sections.

\@@glossarysecstar

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

```

\@@glossaryseclabel

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

```

\glsautoprefix Prefix to add before label if automatically generated:

```
\newcommand*\glsautoprefix}{}

```

numberedsection

```

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

```

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

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

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

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

`\glo@seeautonumberlist`

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

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

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

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

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

`\gls@loadlong`

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

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

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

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

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

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

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

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

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

`\@gls@loadlist`

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

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

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

`\@gls@loadtree`

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

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

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

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

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

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

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

`nopostdot` Boolean option to suppress post description dot

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

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

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

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

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

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

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

`\define@glossaryentrycounter`

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

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

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

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

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

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

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

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

`\define@glossarysubentrycounter`

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

Check if counter already defined.

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

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

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

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

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

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

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

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

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

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

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

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

`\glsprestandardsort`

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

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

`\glo@check@sortallowed`

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

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

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

Store entry information when it's defined.

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

No count register required for standard sort.

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

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

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

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

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

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

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

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

Set standard sort as the default:

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

`\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 glos-
sary and set the sort key.
\def\@gls@setsort##1{%

```

Get the parent, if one exists

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

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

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

Set index information for this entry

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

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

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

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

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

Don't store entry index information.

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

No count register required for standard sort.

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

Don't modify sort value.

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

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

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

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

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

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

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

Don't store entry index information.

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

No count register required for standard sort.

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

Clear sort value.

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

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

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

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

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

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

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

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

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

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

`\glsdefaulttype`

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

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

`\acronymtype`

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

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

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

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

```
\define@boolkey{glossaries.sty}{gls}{acronym}[true]{%
  \ifglsacronym
    \renewcommand{\@gls@do@acronymsdef}{%
      \DeclareAcronymList{acronym}%
      \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
      \renewcommand*\acronymtype{acronym}%
    }
  \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][1]{%
      \printglossary[type=\acronymtype,#1]%
    }
  \fi
}
```

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

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

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

```
\@gls@declareoption{acronyms}{%
  \glsacronymtrue

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

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

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

```

    }%
  }%
}

```

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

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

`\@addtoacronymlists`

```

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

```

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

```

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

```

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

`\glsIfListOfAcronyms`

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

```

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

```

Internal command requires label and list to be expanded:

```

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

```

Not found

```

    #4%
  \else

```

Found

```

    #3%
  \fi
}

```

```

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

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

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

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

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

\glscounter
    \newcommand{\glscounter}{page}

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

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

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

```

nohypertypes

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

`\GlossariesWarning` Prints a warning message.

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

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

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

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

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

`nowarn` Define package option to suppress warnings

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

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

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

`nolangwarn` Suppress warning if language support not found.

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

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

```

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

```

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

```

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

```

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

```

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

```

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

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

`\glssetexpandfield`

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

Sets `field` to always expand.

```

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

```

`\glssetnoexpandfield`

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

Sets `field` to never expand.

```

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

```

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

```
\glssetexpandfield{type}
```

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

```
\glssetnoexpandfield{desc}
```



```

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

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

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

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

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

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

    Sanitizing stuff:

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

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

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

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

```

`\@@gls@noidx@nosanitizesort`

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

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

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

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

```

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

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

```

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

```

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

```

`\@gls@removedoption`

```

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

```

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

```

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

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

```

```

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

```

sanitizesort

```

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

```

Default setting:

```

\glsssanitizesorttrue
\glsssetnoexpandfield{sortvalue}%

```

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

```

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

```

```

    deprecated. Use sanitizesort instead}%
}

```

`sanitize`

```

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

```

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

```

\newif\ifglstranslate

```

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

`\@gls@usetranslator`

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

Set the default value:

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

`\@gls@requested@languages`

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

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

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

indexonlyfirst Set whether to only index on first use.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

shortcuts Define acronym shortcuts.

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

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

```
\newcommand*{\glsorder}{word}
```

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

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

order

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

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

```
\newif\ifglsxindy
```


The default is `makeindex`:

```
\glsxindyfalse
```

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

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

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

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

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

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

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

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

Define key to set the language

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

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

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

Define a key to set the code page.

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

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

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

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

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

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

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

`\@domakeglossaries`

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

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

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

`restoremakegloss` Cancel the effect of `disablemakegloss`.

```
\@gls@declareoption{restoremakegloss}{%
  \ifdefequal\makeglossaries\@no@makeglossaries
  {%
    \GlossariesWarning{Option ‘restoremakegloss’ has no effect
      (\string\makenoidxglossaries\space has already been used)}%
  }%
  {%
    \ifdefequal\@makeglossary\@gobble
    {%
```

```

    \GlossariesWarning{Option 'restoremakeloss' has no effect
    (\string\makeglossaries\space has already been used)}%
  }%
  {%
    \PackageInfo{glossaries}{\string\makeglossaries\space and
    \string\makenoidxglossaries\space have been restored}%
    \let\@domakeglossaries\@firstofone
  }%
}%
}

```

`\do@write@glslabels`

```

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

```

`\@do@write@glslabels`

```

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

```

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

```

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

```

`\@write@glslabelsname`

```

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

```

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

```

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

```

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

```

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

```

`\ifglsautomake`

```

\newif\ifglsautomake

```

`\gls@automake@nr`

```

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

```

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

```

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

```

`\gls@automake@option`

```

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

```

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

```

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

```

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

```

\define@choicekey{glossaries.sty}{automake}%
[\gls@automake@val\gls@automake@nr]{true,false,immediate,makegloss,lite,delayed}[immediate]{%
\ifcase\gls@automake@nr\relax
\GlossariesWarning{automake=true deprecated. Use automake=delayed for a delayed write}%
case 0: true
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\or
case 1: false
\glsautomakefalse
\renewcommand{\gls@automake@makegloss}{}%
\or

```

```

case 2: immediate
    \glsautomaketrue
    \renewcommand{\gls@automake@makegloss}{}%
    \or
case 3: makegloss
    \glsautomaketrue
    \renewcommand{\gls@automake@makegloss}{makeglossaries}%
    \or
case 4: lite
    \glsautomaketrue
    \renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
    \or
case 5: delayed
    \def\gls@automake@nr{0}%
    \glsautomaketrue
    \renewcommand{\gls@automake@makegloss}{}%
    \fi
    \renewcommand{\gls@automake@option}{automake=\gls@automake@val}%
    \ifglsautomake
        \let\gls@doautomake\gls@do@automake@err
    \else
        \renewcommand*\{\gls@doautomake}{}%
    \fi
}
\glsautomakefalse

```

`\gls@doautomake`

```

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

```

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

```

\gls@declareoption{automakegloss}{%
\glsautomaketrue
\renewcommand{\gls@automake@option}{automakegloss}%
\def\gls@automake@val{makegloss}%
\def\gls@automake@nr{3}%
\renewcommand{\gls@automake@makegloss}{makeglossaries}%
\let\gls@doautomake\gls@do@automake@err
}

```

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

```

\gls@declareoption{automakeglosslite}{%
\glsautomaketrue
\renewcommand{\gls@automake@option}{automakeglosslite}%
\def\gls@automake@val{lite}%

```

```

\def\gls@automake@nr{4}%
\renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
\let\@gls@doautomake\@gls@do@automake@err
}

```

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

```

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

```

Set default:

```

\glssavewritesfalse
\let\glswritefiles\@empty

```

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

```

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

```

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

```

\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
\ifbool{glscompatible-2.07}%
{%
\@gls@removedoption{compatible-2.07}%
}%
{}%
}
\boolfalse{glscompatible-2.07}

```

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

```

\let\gls@original@makeglossary\makeglossary

```

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

```

\let\gls@original@glossary\glossary

```

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

`\@gls@override@glossary`

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

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

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

`kernelglossredefs` The `glossaries` package redefines the kernel commands `\makeglossary` and `\glossary` as a legacy action from the former `glossary` package. In hindsight that wasn't a good idea as it's possible that the `glossaries` package may need to be used with another class or package that needs these commands. Neither of these commands are documented in the main user manual and their use is not encouraged. The preferred commands are `\makeglossaries` (to open all associated glossary files) and `\gls`, `\glstext` etc or `\glsadd` for indexing.

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

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

```

\let\makeglossary\gls@original@makeglossary
\or
\def\makeglossary{\makeglossaries}%
\renewcommand*{\@gls@override@glossary}[1][main]{%
\gls@glossary{##1}%
}%
\fi
}

```

symbols Create a “symbols” glossary type

```

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

```

Default is not to define the symbols glossary:

```

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

```

\@gls@symbolsdef

```

\newcommand*{\@gls@symbolsdef}{%
\newglossary[slg]{symbols}{s1s}{s1o}{\glsymbolsgroupname}%
\newcommand*{\printsymbols}[1][\]{\printglossary[type=symbols,##1]}%

```

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

```

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

```

numbers Create a “symbols” glossary type

```

\@gls@declareoption{numbers}{%
\let\@gls@do@numbersdef\@gls@numbersdef
}

```

Default is not to define the numbers glossary:

```

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

```

\@gls@numbersdef

```

\newcommand*{\@gls@numbersdef}{%
\newglossary[nlg]{numbers}{n1s}{n1o}{\glnumbersgroupname}%
\newcommand*{\printnumbers}[1][\]{\printglossary[type=numbers,##1]}%

```

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

```

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

```

index Create an “index” glossary type

```

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

```



```

\fi
}

```

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

```

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

```

Default is not to define index glossary:

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

Now process options passed to the package:

```
\ProcessOptionsX
```

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

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

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

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

Now define `\setupglossaries`:

```
\newcommand*{\setupglossaries}[1]{%
\renewcommand*{\@gls@setacrstyle}{}%
\ifglsacrshortcuts
\def\@gls@setupshortcuts{\glsacrshortcutstrue}%
\else
\def\@gls@setupshortcuts{%
\ifglsacrshortcuts
\DefineAcronymSynonyms
\fi
}%
\fi
\glsacrshortcutsfalse
\let\@gls@do@numbersdef\relax
\let\@gls@do@symbolssdef\relax
\let\@gls@do@indexdef\relax
\let\@gls@do@acronymsdef\relax
\ifglsentrycounter
\let\@gls@doentrycounterdef\relax
\else
\let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
\fi
\ifgls@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
}

```

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

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

```

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

```

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

```

\newcommand*\@gls@onlypremakeg{}

```

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

```

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

```

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

```

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

```

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

```

\newcommand*\@disable@premakecs}[1]{%

```

```

\def#1{\PackageError{glossaries}{\string#1\space may only be
used before \string\makeglossaries}{You can't use
\string#1\space after \string\makeglossaries}}%
}

```

2.3 Predefined Text

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

Main glossary title:

```

\glossaryname
\providecommand*\glossaryname{Glossary}

```

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

```

\acronymname
\providecommand*\acronymname{Acronyms}

```

```

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

```

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

```

\entryname
\providecommand*\entryname{Notation}

```

```

\descriptionname
\providecommand*\descriptionname{Description}

```

```

\symbolname
\providecommand*\symbolname{Symbol}

```

```

\pagelistname
\providecommand*\pagelistname{Page List}

```

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

```

\glssymbolsgroupname
\providecommand*\glssymbolsgroupname{Symbols}

```

```

\glsnumbersgroupname
\providecommand*\glsnumbersgroupname{Numbers}

```

`\glspluralsuffix` The default plural is formed by appending `\glspluralsuffix` to the singular form.

```
\newcommand*\glspluralsuffix}{s}
```

`\glsacrpluralsuffix` Default plural suffix for acronyms

```
\newcommand*\glsacrpluralsuffix}{\glspluralsuffix}
```

`\glsupacrpluralsuffix`

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

`\seename`

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

`\andname`

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

Add multi-lingual support. Thanks to everyone who contributed to the 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.

```
\glsifsettranslator
{%
  \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.

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

Can only be used before `\makeglossaries`

```
\@onlypremakeg\glsSetAlphaCompositor
```

`\gls@suffiX` 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@suffiX{}
```

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

```
\newcommand*\glsSetSuffiX[1]{%
  \renewcommand*\gls@suffiX{#1}}
```

Only has an effect when used before `\makeglossaries`

```
\@onlypremakeg\glsSetSuffiX
```

`\gls@suffiFF` 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@suffiFF{}
```

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

```
\newcommand*\glsSetSuffiFF[1]{%
  \renewcommand*\gls@suffiFF{#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}}%
}

```

`\glslocationstoencap` 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{\glslocationstoencap}[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][\glsdefaulttype]{%
  \ifglossaryexists*{#1}%

```

```

\csgdef{@glossary preamble@#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\glossary preamble{}}
%

```

`\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{\memUHead{#1}}{\memUHead{#1}}%
\else

```

```

        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand{\gls glossarymark}[1]{%
      \ifglsucmark
        \@mkboth{\glsupercase{#1}}{\glsupercase{#1}}%
      \else
        \@mkboth{#1}{#1}%
      \fi
    }
  }
}

```

`\glossarymark` Provided for backward compatibility:

```

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

```

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

`\setglossarysection`

```

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

```

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

`\@glossarysection`

```

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

```

Do automatic labelling if required

```

  \@@glossaryseclabel
}

```

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

`\@p@glossarysection`

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

Do automatic labelling if required

```
\@glossaryseclabel
}
```

`\gls@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 `gls toc` 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*\@glsaddxdyattribute[2]{%
```

Add to xindy attribute list

```

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

```

Add to xindy markup location.

```

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

```

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

```

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

```

High-level command:

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

Add to comma-separated attribute list

```

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

```

First add the attribute without the counter name:

```

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

```

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

```

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

```


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

```
\let\GlsAddXdyCounters\@disabled@glsaddxdycounters
}
```

Only has an effect before `\writeist`:

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

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

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

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

`\@xdyuseralphabets` List of additional alphabets

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

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

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

```

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

```

This code is only required for xindy:

```
\ifglxindy
```

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

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

\@xdyuserlocationnames List of additional user location names
    \def\@xdyuserlocationnames{}

    End of xindy-only block:
    \fi

\@xdycrossrefhook Hook used after writing cross-reference class information.
    \ifglsxindy
    \newcommand\@xdycrossrefhook{}
    \fi

\GlsAddXdyLocation \GlsAddXdyLocation[prefix-loc]{name}{definition} Define a new loca-
tion called name. The definition must use xindy syntax. (Note that this

```

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

```

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

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

```

Only has an effect before `\writeist`:

```

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

```

`\@xdylocationclassorder` Define location class order

```

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

```

Change the location order.

`\GlsSetXdyLocationClassOrder`

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

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

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

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

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

`\GlsSetXdyStyles` Reset the list of required styles

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

```

```

\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.

```

\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

```

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

```

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

```

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

```

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

```

Suggested by egreg:

```

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

```

```

\glsnoxywarning\GlsSetXdyCodePage}
\fi

```

`\@xdylettergroups` Store letter group definitions.

```

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

```

`\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}%
}

```

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

```

\forglentries[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]{cmd}{code}
```

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{type}{true-text}{false-text}
```

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{glo@type@#1@out}{#3}{#2}%
}
```

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

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

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

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

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

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

`\glsdetoklabel`

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

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

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

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

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

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

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

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

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

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

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

`\glsdoifexists`

Generate an error if entry specified by `<label>` doesn't 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@sentries\expandafter
      [\c@name glo@\@gls@thislabel @type\endc@name]
    {\gls@label}%
    {%
      \letcs\gls@parent{gls@\gls@label @parent}%
      \ifdefequal\@gls@thislabel\gls@parent
      {%
        \def\do@glshaschildren{#2}%
        \@endfortrue
      }%
    }%
  }%
  \do@glshaschildren
}%
}
```

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

```
\ifglshasparent
```

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

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

```
\newcommand*{\ifglshasdesc}[3]{%
  \ifcsemtyp{gls@\glsdetoklabel{#1}@desc}%
  {#3}%
  {#2}%
}
```

`\ifglsdescsuppressed` `\ifglsdescsuppressed{<label>}{<true part>}{<false part>}` Does *<true part>* if the description is just `\nopostdesc` otherwise does *<false part>*.

```
\newcommand*\ifglsdescsuppressed}[3]{%
  \ifcsequal{glo\glsdetoklabel{#1}@desc}{@no@post@desc}%
  {#2}%
  {#3}%
}
```

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

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

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

```
\newrobustcmd*\ifglshaslong}[3]{%
  \letcs{\@glo@long}{glo\glsdetoklabel{#1}@long}%
  \ifdefempty\@glo@long
  {#3}%
  {%
    \ifdefequal\@glo@long\@gls@default@value
    {#3}%
    {#2}%
  }%
}
```

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

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

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

`\ifgl$fieldvoid`

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

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

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

\ifglsfield

```
\newrobustcmd*{\ifglsfield}[4]{%
  \glsdoifexists{#2}%
  {%
    \letcs{\@glo@thisvalue}{glo@glstoklabel{#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@glsetoklabel{#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@forbidtextext 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@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.}%
        }%
        {%
        }%
    }
}

```

`\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 glist@#2\endcsname{,}%
```

Store the file extensions:

```
\expandafter\edef\csname @glo@type@#2@log\endcsname{#1}%
\expandafter\edef\csname @glo@type@#2@in\endcsname{#3}%
\expandafter\edef\csname @glo@type@#2@out\endcsname{#4}%
```



```

\expandafter\@gls@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:

```
\ifglxindy
  \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
}%
```

\glsnoidxprenumberlist

```
\newcommand{\glsnoidxprenumberlist}[1]{%
  \cuse{\@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\@gls@detoklabel{##1}@nonumberlist{\@glo@nonumberlist}%
      }%
      {}%
    }%
  \appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}
```

Define some generic user keys. (Additional keys can be added by the user.)

`user1`

```
\define@key{glossentry}{user1}{%
  \def\@glo@useri{#1}%
}
```

`user2`

```
\define@key{glossentry}{user2}{%
  \def\@glo@userii{#1}%
}
```

`user3`

```
\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
```

`user4`

```
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
```

`user5`

```
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
```

user6

```
\define@key{glossentry}{user6}{%  
  \def\@glo@uservi{#1}%  
}
```

short This key is provided for use by `\newacronym`. It's not designed for general purpose use, so isn't described in the user manual.

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

shortplural This key is provided for use by `\newacronym`.

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

long This key is provided for use by `\newacronym`.

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

longplural This key is provided for use by `\newacronym`.

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

`\@glsnoname` Define command to generate error if name key is missing.

```
\newcommand*\@glsnoname{%  
  \PackageError{glossaries}{name key required in  
  \string\newglossaryentry\space for entry '\@glo@label'}{You  
  haven't specified the entry name}}
```

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

```
\newcommand*\@glsnodesc{%  
  \PackageError{glossaries}  
  {%  
    description key required in \string\newglossaryentry\space  
    for entry '\@glo@label'%  
  }%  
  {%  
    You haven't specified the entry description%  
  }%  
}%
```

`\@glsdefaultplural` `\@glsdefaultplural` was obsoleted in v3.10a (2013-10-13) and removed in v4.50.

`\@gls@missingnumberlist` Define a command to generate warning when numberlist not set.

```
\newcommand*\@gls@missingnumberlist}[1]{%
??%
\ifglssavenumberlist
  \GlossariesWarning{Missing number list for entry '#1'.
  Maybe makeglossaries + rerun required}%
\else
  \PackageError{glossaries}%
  {Package option 'savenumberlist=true' required}%
  {%
    You must use the 'savenumberlist' package option
    to reference location lists.%
  }%
\fi
}
```

`\@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
```

`\gls@expandfields` Fully expand values when assigning fields (except for specific fields that are overridden by `\gls@setnoexpandfield`).

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

`\gls@noexpandfields` Don't expand values when assigning fields (except for specific fields that are overridden by `\gls@setexpandfield`).

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```

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

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

```

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

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

```
\AtBeginDocument{\gls@begindocdefs}
```

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

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

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

```

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

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

```

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

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

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

Write key value information:

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

Provide hook:

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

\glswritedefhook

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

\gls@assign@desc

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

\longnewglossaryentry

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

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

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

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

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

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

`\gls@defglossaryentry`

Defines a new entry without checking if it already exists.

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

Prevent any further use of `\GlsSetQuote`:

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

Store label

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

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

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

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

```
\let\@glo@name\@gls@name
\let\@glo@desc\@gls@desc
```

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

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

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

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

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

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

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

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

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

Set the default sort:

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

Set the default counter:

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

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

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

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

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

```
\@gls@initnonnumberlist
```

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

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

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

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

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

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

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

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

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

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

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

```
\@newglossaryentryprehook
```

Extract key-val information from third parameter:

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

Check there is a default glossary.

```
\ifundef\glsdefaulttype
```

```
{%
```

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

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

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

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

```
}%
```

```
{}%
```

Assign type. This must be fully expandable

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

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

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

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

```
{%
```

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

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

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

```
in it}%
```

```
}%
```

```
{%
```

Check if it's an ignored glossary

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

```
{%
```


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

```
\ifx\@glo@desc\@glsnodesc
  \let\@glo@desc\@empty
\fi
}%
{%
}%
\protected@edef\@glo@list@{\csname 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.

```
\ifdequal\@glo@label\@glo@parent%
{%
  \PackageError{glossaries}{Entry ‘\@glo@label’ can't be its own parent}{}%
  \def\@glo@parent{}%
  \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
}%
{%
```

Check the parent exists:

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

Parent exists. Set `\glo@<label>@parent`.

```
\expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
  \@glo@parent}%
```

Determine level.

```
\gls@level=\csname glo@\@glo@parent @level\endcsname\relax
\advance\gls@level by 1\relax
```

If name hasn't been specified, use same as the parent name

```
\ifx\@glo@name\@glsnoname
\expandafter\let\expandafter\@glo@name
\csname glo@\@glo@parent @name\endcsname
```

If name and plural haven't been specified, use same as the parent

```
\ifx\@glo@plural\@gls@default@value
\expandafter\let\expandafter\@glo@plural
\csname glo@\@glo@parent @plural\endcsname
```

```

    \fi
  \fi
}%
{%
```

Parent doesn't exist, so issue an error message and change this entry to have no parent

```

    \PackageError{glossaries}%
    {%
      Invalid parent ‘\@glo@parent’
      for entry ‘\@glo@label’ - parent doesn't exist%
    }%
    {%
      Parent entries must be defined before their children%
    }%
    \def\@glo@parent{}%
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
  }%
}%
\fi
```

Set the level for this entry

```
\expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%
```

Define commands associated with this entry:

```

\gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
\letcs\@glo@sort{glo@\@glo@label @sortvalue}%
\gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @text\endcsname\glspluralsuffix}%
  {\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @text\endcsname}%
  {\@glo@label}{first}{\@glo@first}%
```

If first has been specified, make the default by appending `\glspluralsuffix`, otherwise make the default the value of the plural key.

```

\ifx\@glo@first\@gls@default@value
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @plural\endcsname}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\else
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\fi

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

```

  \def\@glo@defaultcounter{\glscounter}%
}%
```

```

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

```

Set default numberlist if not defined:

```

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

```

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

```

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

```

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

```

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

```

Set the sort key for this entry:

```

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

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

```

```

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

```

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

```

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

```

Sort out any cross-referencing if required.

```
\@glo@autosee
```

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

```

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

```

Define entry counters if enabled:

```
\@newglossaryentry@defcounters
```

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

```

\@newglossaryentryposthook
}

```

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

```

\newcommand*{\@glo@autosee}{%
\ifdefvoid\@glo@see{%
{%
\protected@edef\@do@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{,%}
        \for\glsentries[\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 \@ne
}%
}%
}

\gls@start@measuring It's best to scope this, but still provide some support to switch on and off.
\newcommand*\gls@start@measuring{%
\ifx\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@glsetoklabel{##1}@currcount}{0}\fi
}%
\renewcommand*{\@glslocalreset}[1]{%
  \@glslocalreset{##1}%
  \ifglsresetcurrcount\csdef{glo@glsetoklabel{##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]{%
  \csdef{gls@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{gls@glsdetoklabel{##1}@currcount}{%
    \number\numexpr\glsentrycurrcount{##1}+1}%
  }

```

`\@gls@local@increment@currcount`

```

\newcommand*\@gls@local@increment@currcount}[1]{%
  \csedef{gls@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{s}{\@gls@hyp@opt\@cgl{s}}

```

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

```

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

```

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

```

\def\@cgl{s}@#1#2[#3]{%
  \GlossariesWarning{\string\cgl{s}\space is defaulting to
    \string\gls\space since you haven't enabled entry counting}%
  \@gls@{#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*\cgl{sformat}[2]{%
  \ifglshaslong{#1}{\gl{sentrylong{#1}}{\gl{sentryfirst{#1}}#2}%
}

```

`\cG`ls 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*\cG{l}s{\@gls@hyp@opt\@cG{l}s}

```

Add sentence case mapping.

```

\glsmfuaddmap{\cgl{s}}{\cG{l}s}

```

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

```

\newcommand*\@cG{l}s}[2][{}]{%
  \new@ifnextchar[{\@cG{l}s}@{#1}{#2}}{\@cG{l}s}@{#1}{#2}[]}%
}

```

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

```

\def\@cG{l}s}@#1#2[#3]{%
  \GlossariesWarning{\string\cG{l}s\space is defaulting to
    \string\Gls\space since you haven't enabled entry counting}%
  \@G{l}s@{#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 `\glsformat`. 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}

\glsentryfmt As from version 3.11a, the way in which an entry is displayed is now governed by
\glsentryfmt. This doesn't take any arguments. The required information is
set by commands like \gls. This used to use the now deprecated \glsdisplay
and \glsdisplayfirst. Replaced with \glsgenentryfmt in v4.50.
    \newcommand*\glsentryfmt{\glsgenentryfmt}

\glsgenentryfmt Define a generic format that just uses the first, text, plural or first plural keys
(or the custom text) with the insert text appended.
    \newcommand*\glsgenentryfmt{%
        \ifdefempty\glscustomtext
            {%
                \glsifplural
                    {%
                        Plural form
                            \glscapscase
                                {%
                                    Don't adjust case
                                        \ifglsused\glslabel
                                            {%
                                                Subsequent use
                                                    \glsentryplural{\glslabel}\glsinsert
                                                        }%
                                                    {%
                                                        First use
                                                            \glsentryfirstplural{\glslabel}\glsinsert
                                                                }%
                                                            }%
                                                        {%
                                                            Make first letter upper case
                                                                \ifglsused\glslabel
                                                                    {%
                                                                        Subsequent use.
                                                                            \Glsentryplural{\glslabel}\glsinsert
                                                                                }%
                                                                            {%
                                                                                First use
                                                                                    \Glsentryfirstplural{\glslabel}\glsinsert
                                                                                        }%
                                                                                    }%
                                                                                {%

```

Make all upper case

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

Subsequent use

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

First use

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

Singular form

```
\glscapscase  
{%
```

Don't adjust case

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

Subsequent use

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

First use

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

Make first letter upper case

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

Subsequent use

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

First use

```
\Glsentryfirst{\glslabel}\glsinsert  
}%  
}%  
{%
```

Make all upper case

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

Subsequent use

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

First use

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

Custom text provided in `\glsdisp`. (The insert is most likely to be empty at this point.)

```
\glscustomtext\glsinsert  
}%  
}
```

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

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

Subsequent use:

```
\glsifplural  
}%
```

Subsequent plural form:

```
\glsifplural  
}%
```

Subsequent plural form, don't adjust case:

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

Subsequent plural form, make first letter upper case:

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

Subsequent plural form, all caps:

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

Subsequent singular form

```
\glscapscase  
{%
```

Subsequent singular form, don't adjust case:

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

Subsequent singular form, make first letter upper case:

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

Subsequent singular form, all caps:

```
\glsuppercase  
  {\acronymfont{\glsentryshort{\glslabel}}\glsinsert}%  
}%  
}%  
{%
```

First use:

```
\glsifplural  
{%
```

First use plural form:

```
\glscapscase  
{%
```

First use plural form, don't adjust case:

```
\genplacrfullformat{\glslabel}{\glsinsert}%  
}%  
{%
```

First use plural form, make first letter upper case:

```
\Genplacrfullformat{\glslabel}{\glsinsert}%  
}%  
{%
```

First use plural form, all caps:

```
\glsuppercase  
  {\genplacrfullformat{\glslabel}{\glsinsert}}}%  
}%  
}%  
{%
```

First use singular form

```
\glscapscase  
{%
```

First use singular form, don't adjust case:

```
\genacrfullformat{\glslabel}{\glsinsert}%
```

```
}%  
{%
```

First use singular form, make first letter upper case:

```
\Genacrfullformat{\glslabel}{\glsinsert}%  
}%  
{%
```

First use singular form, all caps:

```
\glsuppercase  
{\genacrfullformat{\glslabel}{\glsinsert}}%  
}%  
}%  
}%  
}%  
{%
```

User supplied text.

```
\glscustomtext  
}%  
}
```

```
\genacrfullformat{<label>}{<insert>}
```

\genacrfullformat

The full format used by \gls`genacfmt` (singular).

```
\newcommand*{\genacrfullformat}[2]{%  
  \glsentrylong{#1}#2\space  
  (\protect\firstacronymfont{\glsentryshort{#1}})%  
}
```

```
\Genacrfullformat{<label>}{<insert>}
```

\Genacrfullformat

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{<label>}{<insert>}
```

\genplacrfullformat

The full format used by \gls`genacfmt` (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}}

```

```

\@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 \@glslink.
    \newcommand*\gls@setdefault@glslink@opts{}

\@gls@link
    \def\@gls@link[#1]#2#3{%
        Inserting \leavevmode suggested by Donald Arseneau (avoids problem with
        tabularx).
        \leavevmode
        \protected@edef\glslabel{\glsdetoklabel{#2}}%
        Save options in \@gls@link@opts and label in \@gls@link@label
        \def\@gls@link@opts{#1}%
        \let\@gls@link@label\glslabel

        \def\@glsnumberformat{glsnumberformat}%
        \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%

        If this is in one of the “nohypertypes” glossaries, suppress the hyperlink by
        default
        \protected@edef\glstype{\csname glo@\glslabel @type\endcsname}%

        Save original setting
        \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper

```

Save original setting of local key.

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

Set defaults:

```
\@gls@setdefault@glslink@opts
```

Switch off hyper setting if the glossary type has been identified in nohyperlist.

```
\do@gls@disablehyperinlist
```

Macros must set this before calling `\@gls@link`. The commands that check the first use flag should set this to `\@gls@link@checkfirsthyper` otherwise it should be set to `\relax`.

```
\do@gls@link@checkfirsthyper  
\setkeys{glslink}{#1}%
```

Add a hook for the user to customise things after the keys have been set.

```
\glslinkpostsetkeys
```

Store the entry's counter in `\theglsentrycounter`

```
\@gls@saveentrycounter
```

Define sort key if necessary:

```
\@gls@setsort{\glslabel}%
```

(De-tok'ing done by `\@do@wrglossary`)

```
\glsencapwrcontent{\@do@wrglossary{#2}}%  
\ifKV@glslink@hyper  
  \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%  
\else  
  \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%  
\fi
```

Restore original setting

```
\let\ifKV@glslink@hyper\org@ifKV@glslink@hyper  
\@gls@restore@glslocal  
}
```

`\glsencapwrcontent` This command is also provided by glossaries-extra v1.49. It's purpose is to resolve [issue #189](#).

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

`\glolinkprefix`

```
\newcommand*{\glolinkprefix}{glo:}
```

`\glsentrycounter` Set default value of entry counter

```
\def\glsentrycounter{\glscounter}%
```

`\@gls@saveentrycounter` Need to check if using equation counter in align environment:

```
\newcommand*{\@gls@saveentrycounter}{%  
  \def\@gls@Hcounter{}}%
```

Are we using equation counter?

```
\ifthenelse{\equal{\@gls@counter}{equation}}%  
{
```

If we're in align environment, `\xatlevel@` will be defined. (Can't test for `\@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}%  
  \fi  
}%
```

Check if hyperref version of this counter

```
\ifcsundef{theH\@gls@counter}%  
{%  
  \def\@gls@Hcounter{\theglsentrycounter}%  
}%  
{%  
  \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%  
}%  
  \protected@edef\theglsentrycounter{\@gls@Hcounter}%  
  \restorecounters@  
  \fi  
}%  
}%  
{%
```

Not using equation counter so no special measures:

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

Check if hyperref version of this counter

```
\ifx\@gls@Hcounter\@empty  
  \ifcsundef{theH\@gls@counter}%  
  {%  
    \def\theglsentrycounter{\theglsentrycounter}%  
  }%  
  {%  
    \protected@edef\theglsentrycounter{\expandafter\noexpand  
      \csname theH\@gls@counter\endcsname}%  
  }%  
  \fi  
}
```

```
}
```

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

```
\def\@set@glo@numformat#1#2#3#4{%  
  \expandafter\@glo@check@mkidxrangear#3\@nil  
  \protected@edef#1{%  
    \@glo@prefix setentrycounter[#4]{#2}%  
    \expandafter\string\csname\@glo@suffix\endcsname  
  }%  
  \@gls@checkmkidxchars#1%  
}
```

Check to see if the given string starts with a (or). If it does set `\@glo@prefix` to the starting character, and `\@glo@suffix` to the rest (or `glsnumberformat` if there is nothing else), otherwise set `\@glo@prefix` to nothing and `\@glo@suffix` to all of it.

```
\def\@glo@check@mkidxrangear#1#2\@nil{%  
\if#1\relax  
  \def\@glo@prefix{()%  
  \if\relax#2\relax  
    \def\@glo@suffix{glsnumberformat}%  
  \else  
    \def\@glo@suffix{#2}%  
  \fi  
\else  
  \if#1)\relax  
    \def\@glo@prefix{)}%  
  \if\relax#2\relax  
    \def\@glo@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{%
```

```

\do@glsexdycheckbackslash\glsexdystring\noexpand@nil
\@backslashchar\@backslashchar\noexpand\null}%
\do@glsexdycheckbackslash
\expandafter\@glsexupdatechecked\@glsexcheckedmkidx{\glsexdystring}%
\def\@glsexcheckedmkidx{}%
\expandafter\@glsexdycheckquote\glsexdystring\@nil""\null
\expandafter\@glsexupdatechecked\@glsexcheckedmkidx{\glsexdystring}%
Unsanitize \glsexnumberpage, \glsexalphpage, \glsexAlphpage and \glsexromanpage
(thanks to David Carlisle for the suggestion.) Skip if esclocations=false.
\ifglsexesclocations
\@for\@glsex@tmp:=\glsexprotected@pagefmts\do
{%
\edef\@glsexsanitized@tmp{\expandafter\@gobble\string\\ \expandonce\@glsex@tmp}%
\@onelevel@sanitize\@glsexsanitized@tmp
\edef\glsex@dosubst{%
\noexpand\DTLsubstituteall\noexpand\glsexdystring
{\@glsexsanitized@tmp}{\expandonce\@glsex@tmp}%
}%
\glsex@dosubst
}%
\fi
Assign to required control sequence
\let#1=\glsexdystring
}

```

Catch special characters (argument must be a control sequence):

```

\@glsexcheckmkidxchars
\newcommand{\@glsexcheckmkidxchars}[1]{%
\ifglsexindy
\@glsexescbsdq{#1}%
\else
\def\@glsexcheckedmkidx{}%
\expandafter\@glsexcheckquote#1\@nil""\null
\expandafter\@glsexupdatechecked\@glsexcheckedmkidx{#1}%
\def\@glsexcheckedmkidx{}%
\expandafter\@glsexcheckescquote#1\@nil\\"\null
\expandafter\@glsexupdatechecked\@glsexcheckedmkidx{#1}%
\def\@glsexcheckedmkidx{}%
\expandafter\@glsexcheckescactual#1\@nil\?\?\null
\expandafter\@glsexupdatechecked\@glsexcheckedmkidx{#1}%
\def\@glsexcheckedmkidx{}%
\expandafter\@glsexcheckactual#1\@nil??\null
\expandafter\@glsexupdatechecked\@glsexcheckedmkidx{#1}%
\def\@glsexcheckedmkidx{}%
\expandafter\@glsexcheckbar#1\@nil||\null
\expandafter\@glsexupdatechecked\@glsexcheckedmkidx{#1}%
\def\@glsexcheckedmkidx{}%
\expandafter\@glsexcheckescbar#1\@nil\\|\null

```

```

\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checklevel#1\@nil!!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%

\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkesclevel#1\@nil!!\!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\edef\@do@gls@checkesc{%
  \noexpand\@gls@checkesc\expandonce{#1}\noexpand\@nil
  \@backslashchar\@backslashchar\noexpand\null
}%
\do@gls@checkesc
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\fi
}

```

Update the control sequence and strip trailing \@nil:

\@gls@updatechecked

```
\def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
```

\@gls@tmpb Define temporary token

```
\newtoks\@gls@tmpb
```

\@gls@checkquote Replace " with "" since " is a makeindex special character.

```

\def\@gls@checkquote#1"#2"#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
  \ifx\null#3\null
  \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
  \def\@gls@checkquote{\relax}%
  \else
  \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
    \@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
  \def\@gls@checkquote{\@gls@checkquote#3\null}%
  \fi
  \else
  \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
    \@gls@quotechar\@gls@quotechar}%
  \ifx\null#3\null
  \def\@gls@checkquote{\@gls@checkquote#2""\null}%
  \else
  \def\@gls@checkquote{\@gls@checkquote#2"#3\null}%
  \fi
  \fi
  \@gls@checkquote
}

```


`\@gls@checkescquote` Do the same for `\`:

```
\def\@gls@checkescquote#1"#2"#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
  \ifx\null#3\null
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
    \def\@@gls@checkescquote{\relax}%
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\string"\@gls@quotechar
      \@gls@quotechar\string"\@gls@quotechar}%
    \def\@@gls@checkescquote{\@gls@checkescquote#3\null}%
  \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\string"\@gls@quotechar}%
    \ifx\null#3\null
      \def\@@gls@checkescquote{\@gls@checkescquote#2"\null}%
    \else
      \def\@@gls@checkescquote{\@gls@checkescquote#2"#3\null}%
    \fi
  \fi
  \@@gls@checkescquote
}
```

`\@gls@checkescactual` Similarly for `\?` (which is replaces `@` as `makeindex`'s special character):

```
\def\@gls@checkescactual#1"?#2"?#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
  \ifx\null#3\null
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
    \def\@@gls@checkescactual{\relax}%
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\string"\@gls@actualchar
      \@gls@quotechar\string"\@gls@actualchar}%
    \def\@@gls@checkescactual{\@gls@checkescactual#3\null}%
  \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\string"\@gls@actualchar}%
    \ifx\null#3\null
      \def\@@gls@checkescactual{\@gls@checkescactual#2"??\null}%
    \else
      \def\@@gls@checkescactual{\@gls@checkescactual#2"?#3\null}%
    \fi
  \fi
  \@@gls@checkescactual
}
```

```
}
```

`\@gls@checkesbar` Similarly for `\|`:

```
\def\@gls@checkesbar#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@checkesbar{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\string"\@gls@encapchar
        \@gls@quotechar\string"\@gls@encapchar}%
      \def\@@gls@checkesbar{\@gls@checkesbar#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\string"\@gls@encapchar}%
    \ifx\null#3\null
      \def\@@gls@checkesbar{\@gls@checkesbar#2\|\|\null}%
    \else
      \def\@@gls@checkesbar{\@gls@checkesbar#2|#3\null}%
    \fi
  \fi
  \@@gls@checkesbar
}
```

`\@gls@checkescllevel` Similarly for `\!`:

```
\def\@gls@checkescllevel#1#!#2#!#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@@gls@checkescllevel{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\string"\@gls@levelchar
        \@gls@quotechar\string"\@gls@levelchar}%
      \def\@@gls@checkescllevel{\@gls@checkescllevel#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\string"\@gls@levelchar}%
    \ifx\null#3\null
      \def\@@gls@checkescllevel{\@gls@checkescllevel#2\!\!\null}%
    \else
      \def\@@gls@checkescllevel{\@gls@checkescllevel#2#!#3\null}%
    \fi
}
```

```

\fi
\@gls@checkesclevel
}

```

\@gls@checkbar and for |:

```

\def\@gls@checkbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@gls@checkbar{\@gls@checkbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkbar{\@gls@checkbar#2||\null}%
\else
\def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}

```

\@gls@checklevel and for !:

```

\def\@gls@checklevel#1!#2!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checklevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
\def\@gls@checklevel{\@gls@checklevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checklevel{\@gls@checklevel#2!!\null}%
\else
\def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
\fi

```

```

\fi
\@@gls@checklevel
}

```

\@gls@checkactual and for ?:

```

\def\@gls@checkactual#1?#2?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
\def\@@gls@checkactual{\@gls@checkactual#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar}%
\ifx\null#3\null
\def\@@gls@checkactual{\@gls@checkactual#2??\null}%
\else
\def\@@gls@checkactual{\@gls@checkactual#2?#3\null}%
\fi
\fi
\@@gls@checkactual
}

```

\@gls@checkesc and for \:

```

\bgroup
\catcode'\|0\relax
\catcode'\|12\relax
\gdef|\@gls@checkesc#1\#2\#3\null{%
|\@gls@tmpb=\expandafter{|\@gls@checkedmkidx}%
\toks@={#1}%
\ifx|null#2|null
\ifx|null#3|null
\edef|\@gls@checkedmkidx{|\the|\@gls@tmpb|\the|\toks@}%
\def|\@@gls@checkesc{|\relax}%
\else
\edef|\@gls@checkedmkidx{|\the|\@gls@tmpb|\the|\toks@
|\@gls@quotechar|\@backslashchar|\@gls@quotechar|\@backslashchar}%
\def|\@@gls@checkesc{|\@gls@checkesc#3|null}%
\fi
\else
\edef|\@gls@checkedmkidx{|\the|\@gls@tmpb|\the|\toks@
|\@gls@quotechar|\@backslashchar}%
\ifx|null#3|null
\def|\@@gls@checkesc{|\@gls@checkesc#2\\|null}%

```

```

        |else
        |def|@gls@checkesc{|@gls@checkesc#2|#3|null}%
        |fi
        |fi
    |@gls@checkesc
}
|egroup

```

`\gls@xdycheckquote` As before but for use with xindy

```

\def@gls@xdycheckquote#1"#2"#3|null{%
  \gls@tmpb=\expandafter{\gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef@gls@checkedmkidx{\the@gls@tmpb\the\toks@}%
      \def@gls@xdycheckquote{\relax}%
    \else
      \edef@gls@checkedmkidx{\the@gls@tmpb\the\toks@
        \string"\string"}%
      \def@gls@xdycheckquote{\gls@xdycheckquote#3|null}%
    \fi
  \else
    \edef@gls@checkedmkidx{\the@gls@tmpb\the\toks@
      \string"}%
    \ifx\null#3\null
      \def@gls@xdycheckquote{\gls@xdycheckquote#2""\null}%
    \else
      \def@gls@xdycheckquote{\gls@xdycheckquote#2"#3\null}%
    \fi
  \fi
  \gls@xdycheckquote
}

```

`\gls@xdycheckbackslash` Need to escape all backslashes for xindy. Define command that will define

```

\gls@xdycheckbackslash
\edef\def@gls@xdycheckbackslash{%
  \noexpand\def\noexpand@gls@xdycheckbackslash##1\@backslashchar
  ##2\@backslashchar##3\noexpand\null{%
    \noexpand@gls@tmpb=\noexpand\expandafter
    {\noexpand@gls@checkedmkidx}%
    \noexpand\toks@={##1}%
    \noexpand\ifx\noexpand\null##2\noexpand\null
    \noexpand\ifx\noexpand\null##3\noexpand\null
    \noexpand\edef\noexpand@gls@checkedmkidx{%
      \noexpand\the\noexpand@gls@tmpb\noexpand\the\noexpand\toks@}%
    \noexpand\def\noexpand@gls@xdycheckbackslash{\relax}%
    \noexpand\else
    \noexpand\edef\noexpand@gls@checkedmkidx{%
      \noexpand\the\noexpand@gls@tmpb\noexpand\the\noexpand\toks@
      \@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
  }

```

```

\noexpand\def\noexpand\@gls@xdycheckbackslash{%
  \noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
  \@backslashchar\@backslashchar}%
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
  \noexpand\@gls@xdycheckbackslash##2\@backslashchar
  \@backslashchar\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
  \noexpand\@gls@xdycheckbackslash##2\@backslashchar
  ##3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@xdycheckbackslash
}%
}
Now go ahead and define \@gls@xdycheckbackslash
\def@gls@xdycheckbackslash

```

`\glsdohypertarget`

```

\newlength@gls@tmplen
\newcommand*\glsdohypertarget}[2]{%
  \glsdoshowtarget{#1}%
  {%
    \glsmeasureheight{gls@tmplen}{#2}%
    \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
}

```

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

```

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

```

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

```

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

```

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

```

\glsenablehyper
\newcommand{\glsenablehyper}{%
\KV@glslink@hypertrue
\let\@glslink\glsdohyperlink
\let\@glsstarget\glsdohypertarget
}

```

Provide some convenience commands if not already defined:

```

\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}

```

Syntax:

```

\gls[<options>]{<label>}[<insert text>]

```

Link to glossary entry using singular form. The link text is taken from the value of the text or first keys used when the entry was defined.

The first optional argument is a key-value list, the same as `\glslink`, the mandatory argument is the entry label. After the mandatory argument, there is another optional argument to insert extra text in the link text (the location of the inserted text is governed by `\glsdisplay` and `\glsdisplayfirst`). As with `\glslink` there is a starred version which is the same as the unstarred

version but with the `hyper` key set to `false`. (Additional options can also be specified in the first optional argument.)

First determine which version is being used:

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

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

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

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

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

    \let\glsifplural\@secondoftwo
    \let\gls caps case\@firstofthree
    \let\gls custom text\@empty
    \def\gls insert{#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 `\acronym type`, suppress hyperlink for first use. Likewise if the `hyperfirst=false` package option is used.

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

Indicate that this entry has now been used

```
\@gls@do@glsunset{#2}%
}%
```

```
\gls post link hook
}
```

`\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.

```
\gls mfu add map{\gls}{\Gls}
```


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

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

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

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

    \let\glsifplural\@secondoftwo
    \let\glsifscaps\@secondofthree
    \let\glsifcustomtext\@empty
    \def\glsinsert{#3}%

```

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

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

Call `\@gls@link` If footnote package option has been used and the glossary type is `\acronymtype`, suppress hyperlink for first use. Likewise if the `hyperfirst=false` package option is used.

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

Indicate that this entry has now been used

```
\@gls@do@glsunset{#2}%
}%
```

```
\glspostlinkhook
}
```

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

`\GLS`

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

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

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

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

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

```

```

\let\glsifplural\@secondoftwo
\let\glsifcaps\@thirdofthree
\let\glsifcustomtext\@empty
\def\glsinsert{#3}%

```

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

```

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

```

Call `\@gls@link`. If footnote package option has been used and the glossary type is `\acronym@type`, suppress hyperlink for first use. Likewise if the `hyperfirst=false` package option is used.

```

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

```

Indicate that this entry has now been used

```

\@gls@do@glssunset{#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\glsifcaps\@firstofthree
\let\glsifcustomtext\@empty
\def\glsinsert{#3}%

```

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

```

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

```

Call `\@gls@link`. If footnote package option has been used and the glossary type is `\acronym@type`, suppress hyperlink for first use. Likewise if the `hyperfirst=false` package option is used.

```

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

```

Indicate that this entry has now been used

```
\@gls@do@glsunset{#2}%
}%

\glspostlinkhook
}
```

`\Glspl` behaves in the same way as `\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\glsescapscase\@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\glsescapscase\@thirdofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
  }
```

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

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

Call `\@gls@link`. If footnote package option has been used and the glossary type is `\acronymtype`, suppress hyperlink for first use. Likewise if the `hyperfirst=false` package option is used.

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

Indicate that this entry has now been used

```
\@gls@do@glsunset{#2}%
}%
```

```
\glspostlinkhook
}
```

`\glsdisp` `\glsdisp[<options>]{<label>}{<text>}` This is like `\gls` except that the link text is provided. This differs from `\glslink` in that it uses `\glsdisplay` or `\glsdisplayfirst` and unsets the first use flag.

First determine if we are using the starred form:

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

Defined the un-starred form.

`\@glsdisp`

```
\newcommand*{\@glsdisp}[3] []{%
  \glsdoifexists{#2}{%
```

```
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
```

```
\let\glsifplural\@secondoftwo  
\let\glsupcase\@firstofthree  
\def\glscustomtext{#3}%  
\def\glsinsert{}%
```

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

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

Call `\@gls@link`. If footnote package option has been used and the glossary type is `\acronym@type`, suppress hyperlink for first use. Likewise if the `hyperfirst=false` package option is used.

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

Indicate that this entry has now been used

```
\@gls@do@glsunset{#2}%  
}%  
  
\glspostlinkhook  
}
```

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

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

`\@Glsdisp`

```
\newcommand*\@Glsdisp[3][{}]{%  
  \@glsdisp[#1]{#2}{\glsentencecase{#3}}%  
}  
\glsmfuaddmap{\@Glsdisp}{\@Glsdisp}
```

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

```
\newcommand*\@gls@link@nocheckfirsthyper{}
```

`\@gls@field@link`

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

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

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

`\glstext`

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

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

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

Read in the final optional argument:

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

`\GLStext` behaves like `\glstext` except the text is converted to uppercase.

`\GLStext`

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

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

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

Read in the final optional argument:

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

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

`\Glstext`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glstext}{\Glstext}
```

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

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

Read in the final optional argument:

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

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

`\glsfirst`

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

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

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

Read in the final optional argument:

```
\def\@glsfirst@#1#2[#3] {%  
  \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%  
}
```

`\Glsfirst` behaves like `\glsfirst` except it displays the first letter in uppercase.

`\Glsfirst`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsfirst}{\Glsfirst}
```

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

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

Read in the final optional argument:

```
\def\@Glsfirst@#1#2[#3] {%  
  \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%  
}
```

`\GLSfirst` behaves like `\Glsfirst` except it displays the text in uppercase.

`\GLSfirst`

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

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

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

Read in the final optional argument:

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

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

`\glsplural`

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

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

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

Read in the final optional argument:

```
\def\@glsplural@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%  
}
```

`\Glsplural` behaves like `\glsplural` except that the first letter is converted to uppercase.

`\Glsplural`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsplural}{\Glsplural}
```

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

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

Read in the final optional argument:

```
\def\@Glsplural@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%  
}
```

`\GLSplural` behaves like `\glsplural` except that the text is converted to uppercase.

`\GLSplural`

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

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

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

Read in the final optional argument:

```
\def\@GLSplural@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryplural{#2}#3}}%  
}
```

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

`\glsfirstplural`

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


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

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

Read in the final optional argument:

```
\def\@glsfirstplural@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%  
}
```

`\Glsfirstplural` behaves like `\glsfirstplural` except that the first letter is converted to uppercase.

`\Glsfirstplural`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsfirstplural}{\Glsfirstplural}
```

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

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

Read in the final optional argument:

```
\def\@GLSfirstplural@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\Glsentryfirstplural{#2}#3}%  
}
```

`\GLSfirstplural` behaves like `\glsfirstplural` except that the link text is converted to uppercase.

`\GLSfirstplural`

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

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

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

Read in the final optional argument:

```
\def\@GLSfirstplural@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryfirstplural{#2}#3}}%  
}
```

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

`\glsname`

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

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

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

Read in the final optional argument:

```
\def\@glsname@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryname{#2}#3}%
}
```

`\Glsname` behaves like `\glsname` except that the first letter is converted to uppercase.

`\Glsname`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsname}{\Glsname}
```

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

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

Read in the final optional argument:

```
\def\@GLSname@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\GLSentryname{#2}#3}%
}
```

`\GLSname` behaves like `\glsname` except that the link text is converted to uppercase.

`\GLSname`

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

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

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

Read in the final optional argument:

```
\def\@GLSname@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryname{#2}#3}}%
}
```

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

`\glsdesc`

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

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

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

Read in the final optional argument:

```
\def\@glsdesc@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentrydesc{#2}#3}%
}
```

`\Glsdesc` behaves like `\glsdesc` except that the first letter is converted to uppercase.

`\Glsdesc`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsdesc}{\Glsdesc}
```

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

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

Read in the final optional argument:

```
\def\@Glsdesc@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%
}
```

`\GLSdesc` behaves like `\glsdesc` except that the link text is converted to uppercase.

`\GLSdesc`

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

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

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

Read in the final optional argument:

```
\def\@GLSdesc@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrydesc{#2}#3}}%
}
```

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

`\glsdescplural`

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

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

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

Read in the final optional argument:

```
\def\@glsdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
```

`\Glsdescplural` behaves like `\glsdescplural` except that the first letter is converted to uppercase.

`\Glsdescplural`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsdescplural}{\Glsdescplural}
```

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

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

Read in the final optional argument:

```
\def\@Glsdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
}
```

`\GLSdescplural` behaves like `\glsdescplural` except that the link text is converted to uppercase.

`\GLSdescplural`

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

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

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

Read in the final optional argument:

```
\def\@GLSdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrydescplural{#2}#3}}%
}
```

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

`\glsymbol`

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

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

```
\newcommand*{\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 first letter is converted to uppercase.

`\Glssymbol`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glssymbol}{\Glssymbol}
```

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

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

Read in the final optional argument:

```
\def\@Glssymbol@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\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}{\glsuppercase{\glstentrysymbol{#2}#3}}%  
}
```

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

`\glsymbolplural`

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

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

```
\newcommand*{\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}{\glstentrysymbolplural{#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}{\GLStentrysymbolplural{#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{\glstentrysymbolplural{#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}{\glstentryuseri{#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}{\GLStentryuseri{#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{\glstentryuseri{#2}#3}%
}
```

`\glsuserii` behaves like `\gls` except it always uses the value given by the `user2` key and it doesn't mark the entry as used.

`\glsuserii`

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

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

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

Read in the final optional argument:

```
\def\@glsuserii@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\glsentryuserii{#2}#3}%  
}
```

\Glsuserii behaves like \glsuserii except that the first letter is converted to uppercase.

\Glsuserii

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsuserii}{\Glsuserii}
```

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

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

Read in the final optional argument:

```
\def\@GLSuserii@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\GLSentryuserii{#2}#3}%  
}
```

\GLSuserii behaves like \glsuserii except that the link text is converted to uppercase.

\GLSuserii

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

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

```
\newcommand*\@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}}%  
}
```

\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}{\glstentryuseriii{#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{\glstentryuseriii{#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*{\@GLSuservi}[2][\%
  \new@ifnextchar[{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2}[]}]}
```

Read in the final optional argument:

```
\def\@GLSuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuserv{#2}#3}%
}
```

\glsuservi behaves like \gls except it always uses the value given by the user6 key and it doesn't mark the entry as used.

\glsuservi

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

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

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

Read in the final optional argument:

```
\def\@glsuservi@#1#2[#3]{%
  \gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
```

\Glsuservi behaves like \glsuservi except that the first letter is converted to uppercase.

\Glsuservi

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsuservi}{\Glsuservi}
```

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

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

Read in the final optional argument:

```
\def\@Glsuservi@#1#2[#3]{%
  \gls@field@link{#1}{#2}{\Glsentryuservi{#2}#3}%
}
```

\GLSuservi behaves like \glsuservi except that the link text is converted to uppercase.

\GLSuservi

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

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

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

Read in the final optional argument:

```
\def\@GLSuservi@#1#2[#3]{%
  \gls@field@link{#1}{#2}{\glsuppercase{\glsentryuservi{#2}#3}}%
}
```

Now deal with acronym related keys. First the short form:

\acrshort

```
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\@ns@acrshort}
```

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

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

Read in the final optional argument:

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

    \let\glsifplural\@secondoftwo
    \let\gls caps case\@firstofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \acronymfont{\glsentryshort{#2}}#3%
    }%
  }
```

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

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

\gls post link hook
}
```

`\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\gls label{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@secondofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
  }
```

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

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

\glspostlinkhook
}
```

`\ACRshort`

```
\newrobustcmd*{\ACRshort}{\@gls@hyp@opt\ns@ACRshort}
\glsmfublocker{\ACRshort}
```

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

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

Read in the final optional argument:

```
\def\@ACRshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%

    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsuppercase{\acronymfont{\glsentryshort{#2}}#3}%
    }%
  }%
}
```

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

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

\glspostlinkhook
}
```

Short plural:

`\acrshortpl`

```
\newrobustcmd*{\acrshortpl}{\@gls@hyp@opt\ns@acrshortpl}
```

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

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

Read in the final optional argument:

```
\def\@acrshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

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

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

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

\glspostlinkhook
}
```

`\Acrshortpl`

```
\newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\ns@Acrshortpl}
```

Add sentence case mapping.

```
\glsmfuaddmap{\Acrshortpl}{\Acrshortpl}
```

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

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

Read in the final optional argument:

```
\def\@Acrshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

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

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

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

```

    \glspostlinkhook
  }

\ACRshortpl
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\ns@ACRshortpl}
\glsmfublocker{\ACRshortpl}

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

Read in the final optional argument:
\def\@ACRshortpl#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\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@glstype @entryfmt\endcsname}%
  }%

  \glspostlinkhook
}

\acrlong
\newrobustcmd*{\acrlong}{\@gls@hyp@opt\ns@acrlong}

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

Read in the final optional argument:
\def\@acrlong#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

```



```

\def\glslabel{#2}%
\let\glsifplural\@secondoftwo
\let\gls caps case\@firstofthree
\let\glsinsert\@empty

```

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

```

\def\gls custom text{%
\gls entry long{#2}#3%
}%

```

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

```

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

\gls post link hook
}

```

`\Acr long`

```

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

```

Add sentence case mapping.

```

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

```

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

```

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

```

Read in the final optional argument:

```

\def\@Acr long#1#2[#3]{%
\gls do if exists{#2}%
{%

\let\do@gls@link@check first hyper\@gls@link@no check first hyper

\def\gls label{#2}%
\let\gls if plural\@secondoftwo
\let\gls caps case\@secondofthree
\let\gls insert\@empty

```

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

```

\def\gls custom text{%
\Gls entry long{#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}
  \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\gls caps case\@thirdofthree
    \let\glsinsert\@empty

Bug fix v4.02 removed \acronymfont from \gls customtext (\acronymfont
only designed for short form).
    \def\gls customtext{%
      \glsupper case{\glsentrylong{#2}#3}%
    }%

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

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

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

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

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

`\Acrlongpl`

```
\newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\@ns@Acrlongpl}
```

Add sentence case mapping.

```
\glsmfuaddmap{\Acrlongpl}{\Acrlongpl}
```

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

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

Read in the final optional argument:

```
\def\@Acrlongpl#1#2[#3]{%  
  \glsdoifexists{#2}%  
  {%
```

```
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
```

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

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

```
\def\glscustomtext{%  
  \Glsentrylongpl{#2}#3%  
}%
```

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

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

```

    \glspostlinkhook
  }

\ACRlongpl
\newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\@ns@ACRlongpl}
\glsmfublocker{\ACRlongpl}

Define the un-starred form. Need to determine if there is a final optional argu-
ment
\newcommand*{\ns@ACRlongpl}[2][ ]{%
  \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2} [ ]}%
}

Read in the final optional argument:
\def\@ACRlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty

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

Call \@gls@link. Note that \@gls@link sets \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]{%
  \glstexpdfstring{\@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@acentryname` Now the behaviour when `\setacronymstyle` is used. The original definition of this command used `mfirstuc` internal commands that are now being deprecated.

```
\newrobustcmd*{\@Gls@acentryname}[1]{%
\ifglshaslong{#1}%
{%
```

Just need to ensure that `\acrnameformat` is expanded, which may cause problems as non-robust semantic commands may expand to content that interferes with sentence-casing.

```
\protected@edef\@glo@text{\@Gls@entry@field{#1}{name}}%
\expandafter\glsentencecase\expandafter{\@glo@text}%
```

```

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

```

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

```

\newcommand*{\@org@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo\@glsdetoklabel{#1}@name}%
\@gls@getbody is defined by mfirstuc (which used to be part of glossaries).
    \expandafter\@gls@getbody\@glo@text{}}\@nil
    \expandafter\ifx\@gls@body\glsentrylong\relax
    \expandafter\Glsentrylong\@gls@rest
  \else
    \expandafter\ifx\@gls@body\glsentryshort\relax
    \expandafter\Glsentryshort\@gls@rest
  \else
    \expandafter\ifx\@gls@body\acronymfont\relax

```

Temporarily make `\glsentryshort` behave like `\Glsentryshort`. (This is on the assumption that the argument of `\acronymfont` is `\glsentryshort {<label>}`, as that's the behaviour of the predefined acronym styles.) This is scoped to localise the effect of the assignment.

```

    {%
      \let\glsentryshort\Glsentryshort
      \@glo@text
    }%
  \else
    \expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
    {%
      \let\glsentryshort\Glsentryshort
      \@glo@text
    }%
  \else
    \expandafter\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` key when the entry was defined). The argument is the label associated with the entry. Note that unless you used `description=false` in the `sanitize` package option you may get unexpected results if the `description` key contained any commands.

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

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsentrydesc}{\Glsentrydesc}
```

Plural form:

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

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsentrydescplural}{\Glsentrydescplural}
```

Get the entry text, as specified by the `text` key when the entry was defined. The argument is the label associated with the entry:

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

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsentrytext}{\Glsentrytext}
```

Get the plural form:

```
\glsentryplural
  \newcommand*\glsentryplural[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}{\Glsentryfirst}
```

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

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

```
\Glsentryfirstplural  
  \newcommand*{\Glsentryfirstplural}[1]{%  
    \@Glsentryfield{#1}{firstpl}%  
  }
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentryfirstplural}{\Glsentryfirstplural}
```

`\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][\glstentrytext{\@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[options]{label}

```

Add a term to the glossary without generating any link text. The optional argument indicates which counter to use, and how to format it (using a key-value list) the second argument is the entry label. Note that *options* only has two keys: `counter` and `format` (the `types` key will be ignored).

`\glsadd`

```

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

```

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

```

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

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

```

Store the entry's counter in `\theglsentrycounter`

```

\@gls@saveentrycounter

```

Define sort key if necessary:

```

\@gls@setsort{#2}%

```

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

```

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

```

Identify `\glsadd` as an exclusion, but this will only work if the optional argument is omitted. If the optional argument is required, the `\glsadd` command with arguments will need to be encapsulated with `\MFUskipunc`. See the `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"}`

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

`\glsildechar` Define `\glsildechar` to make it easier to write a tilde character to a file.
`\edef\glsildechar{\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]{%`
`\glsnoxindywarning\GlsSetXdyFirstLetterAfterDigits}`
`\newcommand*\s@GlsSetXdyFirstLetterAfterDigits}{%`

```

        \@GlsSetXdyFirstLetterAfterDigits
    }
\fi

```

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

```

\ifglxindy
  \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
}
\ifglxindy
  \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

```

`\@glsminrange` Define the minimum number of successive location references to merge into a range.

```

\newcommand*\@glsminrange{2}

```

`\GlsSetXdyMinRangeLength` Set the minimum range length. The value must either be `none` or a positive integer. The glossaries package doesn't check if the argument is valid, that is left to `xindy`.

```

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

```

`\writeist`

```

\ifglxindy

```

Code to use if `xindy` is required.

```

\def\writeist{%

```

Define write register if not already defined

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

Update attributes list

```
\@gls@addpredefinedattributes
```

Open the file.

```
\openout\glswrite=\istfilename
```

Write header comment at the start of the file

```
\write\glswrite{;; xindy style file created by the glossaries
package}%
\write\glswrite{;; for document '\jobname' on
\the\year-\the\month-\the\day}%
```

Specify the required styles

```
\write\glswrite{^^J; required styles^^J}
\@for\@xdystyle:=\@xdyrequiredstyles\do{%
\ifx\@xdystyle\@empty
\else
\protected@write\glswrite{{(require
\string"\@xdystyle.xdy\string")}%
\fi
}%
```

List the allowed attributes (possible values used by the format key)

```
\write\glswrite{^^J%
; list of allowed attributes (number formats)^^J}%
\write\glswrite{{(define-attributes ((\@xdyattributes)))}%
```

Define any additional alphabets

```
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
```

Define location classes.

```
\write\glswrite{^^J; location class definitions^^J}%
```

As from version 3.0, locations are now specified as $\langle Hprefix \rangle \langle number \rangle$, 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 `sealso` 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\glossary preamble}%

```

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[1]{\string\providecommand*%
        \expandafter\string
        \csname glsX\@this@ctr X\@this@attr\endcsname[2]}%
      {%
        \string\setentrycounter
          [\expandafter\@gobble\string\#1]{\@this@ctr}%
        \expandafter\string
        \csname\@this@attr\endcsname
          {\expandafter\@gobble\string\#2}%
      }%
    }%
  }%
}%

```

Add the end part of the open tag and the rest of the markup-index information:

```

\write\glswrite{%
\string\begin
{theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
\space\space:close \string"\glpercentchar\glstildechar n\string
\end{theglossary}\string\glossarypostamble
\glstildechar n\string" ^^J\space\space\space
:tree)}%

```

Specify what to put between letter groups

```

\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\glstildechar n\string")}%

```

Specify what to put between entries

```

\write\glswrite{(markup-indexentry
:open \string"\string\relax \string\glsresetentrylist
\glstildechar n\string")}%

```

Specify how to format entries

```

\write\glswrite{(markup-locclass-list :open
\string"\glsopenbrace\string\glossaryentrynumbers
\glsopenbrace\string\relax\space \string"^^J\space\space\space
:sep \string", \string"
:close \string"\glsclosebrace\glsclosebrace\string")}%

```

Specify how to separate location numbers

```

\write\glswrite{(markup-locref-list

```

```
:sep \string\string\delimN\space\string}}%
```

Specify how to indicate location ranges

```
\write\glswrite{(markup-range  
:sep \string\string\delimR\space\string}}%
```

Specify 2-page and 3-page suffixes, if defined. First, the values must be sanitized to write them explicitly.

```
\@onelevel@sanitize\gls@suffixF  
\@onelevel@sanitize\gls@suffixFF  
  
\ifx\gls@suffixF\@empty  
\else  
  \write\glswrite{(markup-range  
    :close "\gls@suffixF" :length 1 :ignore-end)}}%  
\fi  
\ifx\gls@suffixFF\@empty  
\else  
  \write\glswrite{(markup-range  
    :close "\gls@suffixFF" :length 2 :ignore-end)}}%  
\fi
```

Specify how to format locations.

```
\write\glswrite{^^J; define format to use for locations^^J}%  
\write\glswrite{\@xdylocref}}%
```

Specify how to separate letter groups.

```
\write\glswrite{^^J; define letter group list format^^J}%  
\write\glswrite{(markup-letter-group-list  
:sep \string\string\glsgroupskip\glstildechar n\string}}%
```

Define letter group headings.

```
\write\glswrite{^^J; letter group headings^^J}%  
\write\glswrite{(markup-letter-group  
:open-head \string\string\glsgroupheading  
\glsopenbrace\string^^J\space\space\space  
:close-head \string\glsclosebrace\string}}%
```

Define additional letter groups.

```
\write\glswrite{^^J; additional letter groups^^J}%  
\write\glswrite{\@xdylettergroups}}%
```

Define additional sort rules

```
\write\glswrite{^^J; additional sort rules^^J}  
\write\glswrite{\@xdysortrules}}%
```

Hook for any additional information:

```
\@gls@writeisthook
```

Close the style file

```
\closeout\glswrite
```

Suppress any further calls.

```
\let\writeist\relax
\let\noist@\gls@noist@toolate
}
\else
```

Code to use if `makeindex` is required.

```
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"%}
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax
\ifundef{\glswrite}{\newwrite\glswrite}{\relax
\openout\glswrite=\istfilename
\write\glswrite{\glspercentchar\space makeindex style file
created by the glossaries package}
\write\glswrite{\glspercentchar\space for document
'jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '@gls@actualchar'}
\write\glswrite{encap '@gls@encapchar'}
\write\glswrite{level '@gls@levelchar'}
\write\glswrite{quote '@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
\glossarytoctitle]{\string\glossarytitle}\string
\glossarypreamble\string\n\string\begin{theglossary}\string
\glossaryheader\string\n\string"}
\write\glswrite{postamble \string"\string%\string\n\string
\end{theglossary}\string\glossarypostamble\string\n
\string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
\string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}
\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
\string"\string\relax \string\glsresetentrylist\string\n
\string"}

\write\glswrite{delim_0 \string"\string\{\string
\glossaryentrynumbers\string\}\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string
\glossaryentrynumbers\string\}\string\relax \string"}
\write\glswrite{delim_2 \string"\string\{\string
```



```

    \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_t \string"\string}\string}\string"}
\write\glswrite{delim_n \string"\string"\delimN \string"}
\write\glswrite{delim_r \string"\string"\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
  \string"\string}\string\relax
  \string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"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

```

Hook for any additional information:

```
\@gls@writeisthook
```

Close the file and disable \writeist.

```

\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}
\fi

```

`\GlsSetWriteIstHook` Allow user to append information to the style file.

```

\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook

```

`\@gls@writeisthook`

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

`\GlsSetQuote` Allow user to set the `makeindex` quote character. This is primarily for ngerman users who want to use `makeindex`'s `-g` option.

```

\ifglxsindy
\newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\newcommand*\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}%

```

If German is in use, set the extra makeindex option so `makeglossaries` can pick it up.

```
\@ifpackageloaded{tracklang}%
{%
  \IfTrackedLanguage{german}%
  {%
    \def\@gls@extramakeindexopts{-g}%
  }%
  {}%
}%
{}%
```

Need to redefine `\@gls@checkquote`

```
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkquote####1#####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@checkcactual###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@checkbox###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@checkbox###1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checklevel###1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%

\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesclevel###1\noexpand\@nil
\noexpand\!\noexpand\!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%

\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesc###1\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef%

```

```

\noexpand\def\noexpand\@gls@checkescquote####1%
  \expandonce{\csname#1\endcsname}####2\expandonce{\csname#1\endcsname}%
  ####3\noexpand\null{%
  \noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
  \noexpand\toks@={####1}%
  \noexpand\ifx\noexpand\null####2\noexpand\null
  \noexpand\ifx\noexpand\null####3\noexpand\null
  \noexpand\edef\noexpand\@gls@checkedmkidx{%
    \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
  \noexpand\def\noexpand\@@gls@checkescquote{\noexpand\relax}%
  \noexpand\else
  \noexpand\edef\noexpand\@gls@checkedmkidx{%
    \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
    \noexpand\@gls@quotechar\noexpand\string\expandonce{%
      \csname#1\endcsname}\noexpand\@gls@quotechar
    \noexpand\@gls@quotechar\noexpand\string\expandonce{%
      \csname#1\endcsname}\noexpand\@gls@quotechar}%
  \noexpand\def\noexpand\@@gls@checkescquote{%
    \noexpand\@gls@checkescquote####3\noexpand\null}%
  \noexpand\fi
  \noexpand\else
  \noexpand\edef\noexpand\@gls@checkedmkidx{%
    \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
    \noexpand\@gls@quotechar\noexpand\string
    \expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
  \noexpand\ifx\noexpand\null####3\noexpand\null
  \noexpand\def\noexpand\@@gls@checkescquote{%
    \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
    \expandonce{\csname#1\endcsname}\noexpand\null}%
  \noexpand\else
  \noexpand\def\noexpand\@@gls@checkescquote{%
    \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
    ####3\noexpand\null}%
  \noexpand\fi
  \noexpand\fi
  \noexpand\@@gls@checkescquote
  }%
}%
\@gls@docheckquotedef
}
\newcommand*{\gls@nosetquote}[1]{\PackageError{glossaries}%
  {\string\GlsSetQuote\space not permitted here}%
  {Move \string\GlsSetQuote\space earlier in the preamble, as
  soon as possible after glossaries.sty has been loaded}}
\fi
\@gls@extramakeindexopts
\newcommand*{\@gls@extramakeindexopts}[1]{

```

The command `\noist` will suppress the creation of the `.ist` file. Obviously

you need to use this command before `\writeist` to have any effect.

```
\noist
  \newcommand{\noist}{%
Update attributes list
  \@gls@addpredefinedattributes
  \let\writeist\relax
  }
```

```
\@gls@noist@toolate
  \newcommand{\@gls@noist@toolate}{%
  \GlossariesWarning{too late to use \string\noist. The style file
has already been written}%
  }
```

`\@makeglossary` is an internal command that takes an argument indicating the glossary type. This command will create the glossary file required by `makeindex` for the given glossary type, using the extension supplied by the `<out-ext>` parameter used in `\newglossary` (and it will also activate the `\glossary` command, and create the customized `.ist` `makeindex` style file).

Note that you can't use `\@makeglossary` for only some of the defined glossaries (with just the base `glossaries` package). You either need to have a `\makeglossaries` for all glossaries or none (otherwise you will end up with a situation where `TEX` is trying to write to a non-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.
\ifglsxindy
\newcommand*\@gls@automake}[1]{%
  \ifglossaryexists{#1}
  {%
    \@closegls{#1}%
    \ifdefstring{glsorder}{letter}%
    {\def\@gls@order{-M ord/letorder }}%
    {\let\@gls@order\@empty}%
    \gls@xdy@set@lang{#1}%

    \ifdefvoid\gls@codepage
    {\let\@gls@codepage\@empty}%
    {\def\@gls@codepage{-C \gls@codepage\space}}%
    \edef\@gls@dothiswrite{\noexpand\write18{xindy
      -I xindy
      \@gls@order
      -L \@gls@langmod\space
      -M \gls@istfilebase\space
      \@gls@codepage
      -t \jobname.\csuse{glotype@#1@log}
      -o \jobname.\csuse{glotype@#1@in}
      \jobname.\csuse{glotype@#1@out}}%
    }%
    \@gls@dothiswrite
  }%
  }
\else
\newcommand*\@gls@automake}[1]{%
  \ifglossaryexists{#1}
  {%
    \@closegls{#1}%

```

```

\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-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.

```

\ifglsxindy
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%
\IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-M ord/letorder }}%
{\let\@gls@order\@empty}%

```

Set `\@gls@langmod`:

```

\gls@xdy@set@lang{#1}%

\ifdefvoid\gls@codepage
{\let\@gls@codepage\@empty}%
{\def\@gls@codepage{-C \gls@codepage\space}}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \gls@istfilebase\space
\@gls@codepage
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}}

```

```

        doesn't exist. Rerun may be required}}%
    }%
    {%
    \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
    }%
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%
\IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-1 }}%
{\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@noprintglossary\@makeglossaries@warn@noprintglossary
```

Declare list parser for `\glsdisplaynumberlist`

```
\ifglsavenumberlist  
  \edef\@gls@dodolistparser{\noexpand\DeclareListParser  
    {\noexpand\glsnumlistparser}{\delimN}}%  
  \@gls@dodolistparser  
\fi
```

Prevent user from also using `\makenoidxglossaries`

```
\let\makenoidxglossaries\@no@makeglossaries
```

Prohibit sort key in `printgloss` family:

```
\renewcommand*\@printgloss@setsort}{%  
  \let\@glo@assign@sortkey\@glo@no@assign@sortkey  
}%
```

Check the automake setting:

```
\ifglsautomake
\renewcommand*{\@gls@doautomake}{%
\protected@edef\gls@tmp{\@gls@automake@types}%
\@for\@gls@type:=\gls@tmp\do{%
\ifdefempty{\@gls@type}{}%
{\@gls@automake{\@gls@type}}%
}%
}%
\fi
```

Check the sort setting:

```
\@glo@check@sortallowed\makeglossaries
}%
}
```

Must occur in the preamble:

```
\@onlypreamble{\makeglossaries}
```

`\makeglossaries@warn@noprintglossary`

```
\newcommand\@makeglossaries@warn@noprintglossary{%
\ifdefstring{\@glo@types}{,}%
{%
\GlossariesWarningNoLine{No glossaries have been defined}%
}%
{%
\GlossariesWarningNoLine{No \string\printglossary\space
or \string\printglossaries\space
found. ^^J(Remove \string\makeglossaries\space if you
don't want any glossaries.) ^^JThis document will not
have a glossary}%
}%
}%
```

`\glswrite` The definition of `\glswrite` has now been moved to `\makeglossaries` so that it's only defined if needed.

If `\makeglossaries` hasn't been used, issue a warning. Also issue a warning if neither `\printglossaries` nor `\printglossary` have been used.

```
\AtEndDocument{%
\warn@nomakeglossaries
\warn@noprintglossary
}
```

`\makenoidxglossaries` Analogous to `\makeglossaries` this activates the commands needed for `\printnoidxglossary`

```
\newcommand*\makenoidxglossaries{%
\def\glsindexingsetting{noidx}%
\@domakeglossaries
}%
```

Redefine empty glossary warning:

```
\renewcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{Empty glossary for
  \string\printnoidxglossary[type={##1}].
  Rerun may be required (or you may have forgotten to use
  commands like \string\gls)}%
}%
```

Don't escape makeindex/xindy characters:

```
\let\@gls@checkmkidxchars\@gobble
```

Don't escape locations:

```
\glsesclocationsfalse
```

Write glossary information to aux instead of glossary files

```
\let\@do@wrglossary\gls@noidxglossary
```

Switch on group headings that use the character code:

```
\let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
```

Allow see key:

```
\let\gls@checkseeallowed\relax
```

Redefine cross-referencing macro:

```
\renewcommand{\@do@seeglossary}[2]{%
  \protected@edef\@gls@label{\glsdetoklabel{##1}}%
  \protected@write\@auxout{}{%
    \string\@gls@reference
    {\csname glo@\@gls@label @type\endcsname}%
    {\@gls@label}%
    {%
      \string\glsseeformat##2}%
    }%
  }%
}%
```

If user removes the glossaries package from their document, ensure the next run doesn't throw a load of undefined control sequence errors when the aux file is parsed.

```
\AtBeginDocument
{%
  \write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
}%
```

Change warning about no glossaries

```
\def\warn@noprintglossary{%
  \GlossariesWarningNoLine{No \string\printnoidxglossary\space
  or \string\printnoidxglossaries ^^J
  found. (Remove \string\makenoidxglossaries\space if you
  don't want any glossaries.)^^JThis document will not have a glossary}%
}%
```

```

Suppress warning about no \makeglossaries
\let\warn@nomakeglossaries\relax
Prevent user from also using \makeglossaries
\let\makeglossaries\@no@makeglossaries
Allow sort key in printgloss family:
\renewcommand*{\@printgloss@setsort}{%
\let\@glo@assign@sortkey\@glo@assign@sortkey
Initialise default sort order:
\def\@glo@sorttype{\@glo@default@sorttype}%
}%
All entries must be defined in the preamble:
\renewcommand*\new@glossaryentry[2]{%
\PackageError{glossaries}{Glossary entries must be
defined in the preamble^^Jwhen you use
\string\makenoidxglossaries}%
{Either move your definitions to the preamble or use
\string\makeglossaries}%
}%
Redefine \glsentrynumberlist
\renewcommand*\glsentrynumberlist[1]{%
\letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
\ifdef\@gls@loclist
{%
\glsnoidxloclist{\@gls@loclist}%
}%
{%
??\glsdoifexists{##1}%
{%
\GlossariesWarning{Missing location list for ‘##1’. Either
a rerun is required or you haven’t referenced the entry}%
}%
}%
}%
Redefine \glsdisplaynumberlist
\renewcommand*\glsdisplaynumberlist[1]{%
\letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
\ifdef\@gls@loclist
{%
\def\@gls@noidxloclist@sep{%
\def\@gls@noidxloclist@sep{%
\def\@gls@noidxloclist@sep{%
\glsnumlistsep
}%
\def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
}%
}%
}%

```

```

\def\@gls@noidxloclist@finalsep{}%
\def\@gls@noidxloclist@prev{}%
\forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
\@gls@noidxloclist@finalsep
\@gls@noidxloclist@prev
}%
{%
??\glsdoifexists{##1}%
{%
\GlossariesWarning{Missing location list for ‘##1’. Either
a rerun is required or you haven’t referenced the entry}%
}%
}%
}%

```

Provide a generic way of iterating through the number list:

```

\renewcommand*{\glsnumberlistloop}[3]{%
\letcs{\@gls@loclist}{glo\@glsdetoklabel{##1}@loclist}%
\let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
\let\@gls@org@glsseeformat\glsseeformat
\let\glsnoidxdisplayloc##2\relax
\let\glsseeformat##3\relax
\ifdef\@gls@loclist
{%
\forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
}%
{%
??\glsdoifexists{##1}%
{%
\GlossariesWarning{Missing location list for ‘##1’. Either
a rerun is required or you haven’t referenced the entry}%
}%
}%
\let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
\let\glsseeformat\@gls@org@glsseeformat
}%

```

Modify sanitize sort function

```

\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort

```

Check sort option allowed.

```

\@glo@check@sortallowed\makenoidxglossaries

```

Implement nonumberlist

```

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

```

Preamble-only command:

```

\@onlypreamble{\makenoidxglossaries}

```

```
\glsnumberlistloop{<label>}{<handler>}{<xr handler cs>}
```

`\glsnumberlistloop`

```
\newcommand*\glsnumberlistloop[3]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
```

`\glsnoidxnumberlistloophandler` Handler macro for `\glsnumberlistloop`. (The argument should be in the form `\glsnoidxdisplayloc{<prefix>}{<counter>}{<format>}{<n>}`)

```
\newcommand*\glsnoidxnumberlistloophandler[1]{%
  #1%
}
```

`\@no@makeglossaries` Can't use both `\makeglossaries` and `\makenoidxglossaries`

```
\newcommand*\@no@makeglossaries{%
  \PackageError{glossaries}{You can't use both
  \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
  together.}%
}
```

`\@gls@noref@warn` Warning when no instances of `\@gls@reference` found.

```
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
    is required to make \string\printnoidxglossary[type={#1}] work}%
}
```

2.14 Writing information to associated files

`\gls@noidxglossary` Write the glossary information to the aux file (for the 'noidx' method).

```
\newcommand*\gls@noidxglossary{%
  \protected@write\@auxout{\glswrglossdisablelocationcmds}{%
    \string\@gls@reference
      {\csname glo@\@gls@label @type\endcsname}%
      {\@gls@label}%
      {\string\glsnoidxdisplayloc
        {\@glo@counterprefix}%
        {\@gls@counter}%
        {\@glsnumberformat}%
        {\@glslocref}%
      }%
    }%
}
```

`\istfile` Deprecated in v3.0 (2011-04-02). Removed in v4.50.

At the end of the document, the files should be created if `savewrites=true`.

```
\AtEndDocument{%  
  \glswritefiles  
}
```

`\@glswritefiles` Only write the files if `savewrites=true`.

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

Iterate through all the glossaries.

```
\forallglossaries{\@glo@type}{%
```

Check for empty glossaries (patch provided by Patrick Häcker)

```
\ifcsundef{glo@\@glo@type @filetok}%  
  {%  
    \def\gls@tmp{}%  
  }%  
  {%  
    \edef\gls@tmp{\expandafter\the  
      \csname glo@\@glo@type @filetok\endcsname}%  
  }%  
  \ifx\gls@tmp\@empty  
    \ifx\@glo@type\glsdefaulttype  
      \GlossariesWarningNoLine{Glossary ‘\@glo@type’ has no  
        entries.^^JRemember to use package option ‘nomain’ if  
you  
        don’t want to^^Juse the main glossary}%  
    \else  
      \GlossariesWarningNoLine{Glossary ‘\@glo@type’ has no  
        entries}%  
    \fi  
  \else  
    \@glsopenfile{\glswrite}{\@glo@type}%  
    \immediate\write\glswrite{%  
      \expandafter\the  
        \csname glo@\@glo@type @filetok\endcsname}%  
    \immediate\closeout\glswrite  
  \fi  
}%  
}
```

As from v4.10, the `\glossary` command isn't used by the `glossaries` package. Since the user isn't expected to use this command (as `glossaries` takes care of the particular format required for `makeindex/xindy`) there's no need for a user level command. Using a custom internal command prevents any conflict with other packages (and with the `\mark` mechanism).

The associated number should be stored in `\theglsentrycounter` before using `\gls@glossary`.

```
\gls@glossary
```

```
\newcommand*{\gls@glossary}[1]{%
```



```
\@gls@glossary{#1}%
}
```

```
\@gls@glossary{<type>}{<indexing info>}
```

```
\@gls@glossary
```

(In v4.10, `\@glossary` was redefined to `\@gls@glossary` to avoid conflict with other packages.) Initially define internal `\@gls@glossary` to ignore its argument. Indexing will be enabled when `\@gls@glossary` is redefined by `\@makeglossary`.

This command was originally defined to do `\@index{<indexing info>}` so that it behaved much like `\index`. The definition was then changed to use `\index` as `memoir` changes the definition of `\@index`. (Thanks to Dan Luecking for pointing this out.)

However, if normal indexing is enabled (for example with `\makeindex`) but no glossary lists are required (so `\@makeglossary` isn't used), then `\index` will cause a problem here. The `\@index` trick allows for special characters within `<indexing info>` (so you can do, for example, `\index{%\%}`), and the original design of `\@glossary` here was actually a legacy from the old `glossary` package. With the `glossaries` package, the indexing information supplied in the second argument is more constrained and just consists of the sort value (given by the sort key), the actual value (given by `\glossentry{<label>}` or `\subglossentry{<level>}{<label>}`), and the format. This means that there's no need to worry about special characters appearing in the second argument as they can't be in the label or sort value. (If they are in the sort value then the category code would've needed to be changed when the entry was defined or `\glspercentchar` would be needed with the sort sanitization switched off.) This means that it's safe to simply ignore the second argument.

```
\newcommand*{\@gls@glossary}[2]{%
  \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\begin@group\gls@wrglossary{##1}}%
  \let\@gls@renewglossary\@empty
}
```

The `\gls@wrglossary` command is defined to have two arguments. The first argument is the glossary type, the second argument is the glossary entry (the format of which is set in `\glslink`).

```

\gls@wrglossary
\newcommand*{\gls@wrglossary}[2]{%
  \ifglssavewrites
    \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
    \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
    \expandafter{\@gls@tmp^^J}%
  \else
    \ifcsdef{glo@#1@file}%
    {%
      \expandafter\protected@write\csname glo@#1@file\endcsname{%
        \gls@disablepagerefexpansion}{#2}%
    }%
    {%
      \ifignoredglossary{#1}{}%
      {%
        \GlossariesWarning{No file defined for glossary ‘#1’}%
      }%
    }%
  \fi
\endgroup\@esphack
}

```

```

\@do@wrglossary
\newcommand*{\@do@wrglossary}[1]{%
  \gls@ifnotmeasuring{\glswriteentry{#1}{\@do@wrglossary{#1}}}%
}

```

`\glswriteentry` Provide a user level command so the user can customize whether or not a line should be added to the glossary. The arguments are the label and the code that writes to the glossary file.

```

\newcommand*{\glswriteentry}[2]{%
  \ifglsindexonlyfirst
    \ifglsused{#1}{#2}%
  \else
    #2%
  \fi
}

```

`\gls@protected@pagefmts` List of page formats to be protected against expansion.

```

\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}

```

`\gls@disablepagerefexpansion` As from v4.50, this will only do the page protection formats for `esclocations=true`.

```

\newcommand*{\gls@disablepagerefexpansion}{%
  \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 ($\backslash\langle csname \rangle\c@page$ 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@gls#1\endcsname}%
      \expandonce{\csname#1\endcsname}%
      \noexpand\def\expandonce{\csname#1\endcsname}{%
        \noexpand\@wrglossary@pageformat
          \expandonce{\csname gls#1page\endcsname}%
          \expandonce{\csname org@gls#1\endcsname}%
        }%
      }%
    \else
      \GlsWarnAddProtectedPageFmt
    \fi
}

```

```

\GlsWarnAddProtectedPageFmt
    \newcommand{\GlsWarnAddProtectedPageFmt}{%
      \GlossariesWarning{\string\glsaddprotectedpagefmt\space has no
        effect with esclocations=false}%
    }

\@wrglossarynumberhook Hook used by \@do@wrglossary
    \newcommand*\@wrglossarynumberhook{}

\@wrglossary@pageformat
    \newcommand{\@wrglossary@pageformat}[3]{%
      \ifx#3\c@page #1\else #2#3\fi
    }

\@do@wrglossary Write the glossary entry in the appropriate format.
    \newcommand*\@do@wrglossary}[1]{%
      \ifglseclocations
        \@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{\theglslentrycounter}%
      \expandafter\def\expandafter\@glsHlocref\expandafter{\theHglslentrycounter}%
      Find the prefix if \@glsHlocref and \@glslocref aren't the same.
      \ifx\@glsHlocref\@glslocref
        \def\@glo@counterprefix{}%
      \else

```

The value of the counter isn't important here as it's the prefix that's of interest. (`\c@page` will have the same value in both `\theglsentrycounter` and `\theHglentrycounter` at this point, even if it hasn't been updated yet. The page number is not expected to occur in the prefix.)

```
\xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
\fi
```

De-tok label if required.

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

Write the information to file:

```
\@do@wrglossary
}
```

`\ifglswrallowprimitivemods` Conditional to determine whether or not `\@do@esc@wrglossary` should be allowed to temporarily redefine `\the` and `\number`. Version 4.50 switched the default to false.

```
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodsfalse
```

`\@do@esc@wrglossary` Write the glossary entry in the appropriate format. (Need to set `\@glsnumberformat` and `\@gls@counter` prior to use.) The argument is the entry's label. This is far more complicated with `xindy` than with other indexing methods. There are two necessary but conflicting requirements with `xindy`:

1. all backslashes in the location must be escaped;
2. `\c@page` can't be prematurely expanded.

(With `makeindex` there's the remote possibility that the page compositor is a `makeindex` special character, so that would also need to be escaped.)

For example, suppose `\thepage` is defined as

```
%\renewcommand{\thepage}{\tally{page}}
%\newcommand{\tally}[1]{\tallynum{\expandafter\the\csname c@#1\endcsname}}
%
```

where `\tallynum` is a robust command that takes a number as its argument. With all indexing methods other than `xindy`, a deferred write with `\thepage` as the location will expand to `\tallynum{<n>}` where `<n>` is the page number. Since the write is deferred, the page number is correct. (`makeindex` won't accept this location format, but `\makenoidxglossaries` and `bib2gls` are quite happy with it.) Unfortunately, this fails with `xindy` because `xindy` interprets this location as `\tallynum{<n>}` because `\t` represents a the character "t". The location must be written as `\\tallynum{<n>}`.

This means that the location `\tally{page}` must be expanded and then the backslashes must be doubled. Unfortunately `\c@page` mustn't be expanded until the deferred write is performed, so the location actually needs to be expanded to `\tallynum{\the\c@page}` but the backslashes in `\the\c@page` mustn't be

escaped. All other backslashes must be escaped. (In this case, only the backslash in `\tallynum` but the location format may include other control sequences.) The code below works on the assumption that commands like `\tally` are defined in the form

```
%\newcommand{\tally}[1]{\tallynum{\expandafter\the\csname c@#1\endcsname}}
%
```

(note the use of `\expandafter` and `\name`) or in the form

```
%\newcommand{\tally}[1]{\tallynum{\arabic{#1}}}}
%
```

In the second case, `\arabic` is one of the known commands that's temporarily adjusted to prevent `\c@page` from being prematurely expanded. In the first case, `\the` is temporarily modified (unless `\glswrallowprimitivemodsfalse`) to check if it's followed by `\c@page`. The `\expandafter` ensures that it is. If `\tally` is defined in another way that hides `\c@page` for example using `\the\value{#1}` then the process fails.

With `makeindex`, `\tallynum` needs to expand to just the decimal number while writing the location to the glossary file, otherwise `makeindex` will reject it. This can be done by defining `\glstallypage` so that `\tally` can locally be set to `\arabic` while expansion is occurring. Again, `\c@page` must be protected from expansion until the deferred write occurs.

The expansion before the write occurs also allows the hyper prefix to be determined where `\theH<counter>` is defined in the form `<prefix>.\the<counter>`. It's possible (although again unlikely) that a `makeindex` character might occur in the prefix, which therefore needs escaping. The prefix is passed as the optional argument of `\setentrycounter` which is needed by commands like `\glshypernumber` to create a hyperlink for a given counter (like `\hyperpage` but for an arbitrary counter).

```
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
  \begingroup
```

First a bit of hackery to prevent premature expansion of `\c@page`. Store original definitions (scoped):

```
\let\gls@orgthe\the
\let\gls@orgnumber\number
\let\gls@orgarabic\@arabic
\let\gls@orgromannumeral\romannumeral
\let\gls@orgalph\@alph
\let\gls@orgAlph\@Alph
\let\gls@orgRoman\@Roman
```

Redefine:

```
\ifglswrallowprimitivemods
```

The redefinition of `\the` to use `\expandafter` solves the problem of `\the\csname c@<counter>\endcsname` but is only a partial solution to the problem of `\the\value`. With `\value`,

`\c@page` is too deeply hidden and will be expanded too soon, but at least there won't be an error.

```

\def\gls@the##1{%
  \ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
\def\the{\expandafter\gls@the}%
\def\gls@number##1{%
  \ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
\def\number{\expandafter\gls@number}%

\def\romannumeral##1{%
  \ifx##1\c@page \gls@romanpage \else \gls@orgromannumeral##1\fi}%
\fi
\def\@arabic##1{%
  \ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\@Roman##1{%
  \ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
  \ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
  \ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%

```

Add hook to allow for other number formats:

```
\@wrglossarynumberhook
```

Prevent expansion:

```
\gls@disablepagerefexpansion
```

Now store location in `\@glslocref`:

```

\protected@xdef\@glslocref{\theglsentrycounter}%
\endgroup

```

Escape any special characters. It's possible that with `makeindex` the separator might be a `makeindex` special character. Although not likely, it still needs to be taken into account.

```
\@gls@checkmkidxchars\@glslocref
```

Check if the hyper-location is the same as the location and set the hyper prefix.

```

\expandafter\ifx\theHglsentrycounter\theglsentrycounter\relax
  \def\@glo@counterprefix{}%
\else
  \protected@edef\@glsHlocref{\theHglsentrycounter}%
  \@gls@checkmkidxchars\@glsHlocref
  \@xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
\fi

```

De-tok label if required

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

Write the information to file:

```

\@do@wrglossary
}

```

`\@do@wrglossary`

```
\newcommand*{\@do@wrglossary}{%
```

Determine whether to use `xindy` or `makeindex` syntax

```
\ifglxindy
```

Need to determine if the formatting information starts with a (or) indicating a range.

```
\expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
\def\@glo@range{}%
\expandafter\if\@glo@prefix\relax
\def\@glo@range{:open-range}%
\else
\expandafter\if\@glo@prefix\relax
\def\@glo@range{:close-range}%
\fi
\fi
```

Write to the glossary file using `xindy` syntax.

```
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
(indexentry :tkey (\csname glo@\@gls@label @index\endcsname)

:locref \string"\@glo@counterprefix}{\@gls@locref}\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}{\@gls@locref}}%
\fi
}
```

`\@xp@gls@getcounterprefix`

```
\newcommand*\@xp@gls@getcounterprefix[2]{%
\bgrou
\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 location ‘#1’. You need to modify the
definition of \string\theH\@gls@counter otherwise you
will get the warning: “name{\@gls@counter.#1}’ has been
referenced but does not exist”}%
\fi
\fi
}
```

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]{\gls hyperlink[\glsseeitemformat{#1}]{#1}}
```

`\glsseeitemformat` Introduced in v3.0 to replace use of `\glsentryname` (originally in `\glsseeitem`) with `\glsentrytext` to avoid problems with the name key being sanitized. This ceased being a problem in v3.08a, but `\glsentrytext` was retained for backward-compatibility. As from v4.50, the default has reverted back to using `\glsentryname` unless the entry is an acronym (which could have a potentially long name).

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

```
\ifglshashshort{#1}{\glsentrytext{#1}}{\glsentryname{#1}}%
```

```
}
```

2.16 Displaying the glossary

An individual glossary is displayed in the text using `\printglossary[<key-val list>]`. If the `type` key is omitted, the default glossary is displayed. The optional argument can be used to specify an alternative glossary, and can also be used to set the style, title and entry in the table of contents. Available keys are defined below.

`\gls@save@numberlist` Provide command to store number list.

```
\newcommand*\gls@save@numberlist[1]{%
```

```
\ifglssavenumberlist
```

```
\toks@{#1}%
```

```
\protected@edef\@do@writeaux@info{%
```

```
\noexpand\csgdef{glo@\gls@currententrylabel @numberlist}{\the\toks@}%
```

```
}%
```

```
\@onelevel@sanitize\@do@writeaux@info
```

```
\protected@write\@auxout{}{\@do@writeaux@info}%
```

```
\fi
```

```
}
```

`\warn@noprntglossary` 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@noprntglossary}{}
```

`\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=glsdefaulttype]{%
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\@printglossary{#1}\@print@glossary}%
}
```

The `\printglossaries` command will do `\printglossary` for each glossary type that has been defined. It is better to use `\printglossaries` rather than individual `\printglossary` commands to ensure that you don't forget any new glossaries you may have created. It also makes it easier to chop and change the value of the `acronym` package option. However, if you want to list the glossaries in a different order, or if you want to set the title or table of contents entry, or if you want to use different glossary styles for each glossary, you will need to use `\printglossary` explicitly for each glossary type.

```
\printglossaries
  \newcommand*\printglossaries}{%
    \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
  }
```

```
\printnoidxglossary Provide an alternative to \printglossary that doesn't require an external indexing application. Entries won't be sorted and the location list will be empty.
  \newcommand*\printnoidxglossary}[1][type=glsdefaulttype]{%
    \let\@printgloss@checkexists\@printgloss@checkexists@noignored
    \@printglossary{#1}{\@print@noidx@glossary}%
  }
```

```
\printnoidxglossaries Analogous to \printglossaries
  \newcommand*\printnoidxglossaries}{%
    \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
  }
```

```
\@printgloss@setsort Initialise to do nothing.
  \newcommand*\@printgloss@setsort}{}
```

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

```
\@printglossary Sets up the glossary for either \printglossary or \printnoidxglossary. The first argument is the options list, the second argument is the handler macro that deals with the actual glossary. This is also used by glossaries-extra's \printunsrtglossary which may be used with an ignored glossary.
```

```
\newcommand{\@printglossary}[2]{%
Set up defaults.
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{\csname @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@dotocitle{\glssettocitle{\@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 @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@dotocitle

```

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 @glo@type@\@glo@type @in\endcsname}%

```

If the glossary file doesn't exist, do `\null`. (This ensures that the page is shipped out and all write commands are done.) This might produce an empty page, but at this point the document isn't complete, so it shouldn't matter.

```

\IfFileExists{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%
{}%
{\null}%

```

If `xindy` is being used, need to write the language dependent information to the `.aux` file for `makeglossaries`.

```

\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\@xdy@language[2]{}%
    \noexpand\immediate\noexpand\write\@auxout{%
      \string\@xdy@language{\@glo@type}{\@gls@langmod}}%
  }%
}%
\@do@auxoutstuff
\edef\@do@auxoutstuff{%
  \noexpand\AtEndDocument{%
```

If the user removes the glossaries package from their document, ensure the next run doesn't throw a load of undefined control sequence errors when the aux file is parsed.

```
    \noexpand\immediate\noexpand\write\@auxout{%
      \string\providecommand\string\@gls@codepage[2]{}%
    \noexpand\immediate\noexpand\write\@auxout{%
      \string\@gls@codepage{\@glo@type}{\@gls@codepage}}%
  }%
}%
\@do@auxoutstuff
\fi
```

Activate warning if `\makeglossaries` hasn't been used.

```
\renewcommand*\@warn@nomakeglossaries{%
  \GlossariesWarningNoLine{\string\makeglossaries\space
    hasn't been used,^^Jthe glossaries will not be updated}%
}%
}
```

The sort macros all have the syntax:

```
\@glo@sortmacro@<order>{<type>}
```

where `<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{<type>}{<parent>}
```

\@glo@addchildren

```
\newcommand*{\@glo@addchildren}[2]{%
```

Scope to allow nesting.

```

\bgroup
\letcs{\@glo@childlist}{@glo@sortingchildren@#2}%
\@for\@this@childlabel:=\@glo@childlist\do
{%

```

Check this label hasn't already been added.

```

\ifinlistcs{\@this@childlabel}{@glsref@#1}%
{%
%
\listcsxadd{@glsref@#1}{\@this@childlabel}%
}%

```

Does this child have children?

```

\ifcsdef{@glo@sortingchildren@\@this@childlabel}%
{%
\@glo@addchildren{#1}{\@this@childlabel}%
}%
}%
}%
\egroup
}

```

\@glo@do@sortentries

```
\newcommand*{\@glo@do@sortentries}[1]{%
```

```

\ifglshasparent{#1}%
{%
This entry has a parent, so add it to the child list
\protected@edef\@glo@parent{\csuse{glo@\glsdetoklabel{#1}@parent}}%
\ifcsundef{@glo@sortingchildren@\@glo@parent}%
{%
\csdef{@glo@sortingchildren@\@glo@parent}{}%
}%
}%
\expandafter\@glo@sortedinsert
\csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
Has the parent been added?
\xifinlistcs{\@glo@parent}{\glsref@\@glo@type}%
{%
Yes, it has so do nothing.
}%
{%
No, it hasn't so add it now.
\expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
}%
}%
\@glo@sortedinsert{\@glo@sortinglist}{#1}%
}%
}

```

```
\@glo@sortedinsert{\list}{entry label}
```

\@glo@sortedinsert

Insert into list.

```

\newcommand*{\@glo@sortedinsert}[2]{%
\dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
}%

```

The sort handlers need to be in the form required by datatool's `\dtl@sortlist` macro. These must set the count register `\dtl@sortresult` to either -1 ($\#1$ less than $\#2$), 0 ($\#1 = \#2$) or $+1$ ($\#1$ greater than $\#2$).

\@glo@sorthandler@word

```

\newcommand*{\@glo@sorthandler@word}[2]{%
\letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
\letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
\edef\@glo@do@compare{%
\noexpand\dtl@wordindexcompare{\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@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]}{}}%
  }%
}

```

`\@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@ref@#1}%
    {%
      \listcsadd{\@glo@sortinglist}{\@gl@thislabel}%
    }%
    {%

```

Hasn't been referenced.

```

    }%
  }%
  \cslet{\@gl@ref@#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@ref@\@glo@type}%
  {}%
  {%
    \listcsadd{\@gl@ref@\@glo@type}{#1}%
  }%
  \ifcsdef{\@glo@sortingchildren@#1}%
  {%
    \@glo@addchildren{\@glo@type}{#1}%
  }%
  {}%
}

```

`\@glo@sortmacro@use` Sort macro for ‘use’. (No sorting is required, as the entries are already in order of use, so do nothing.)

```

\newcommand*{\@glo@sortmacro@use}[1]{}

```

`\@print@noidx@glossary` Glossary handler for `\printnoidxglossary` which doesn't use an indexing application. Since `\printnoidxglossary` may occur at the start of the document, we can't just check if an entry has been used. Instead, the first pass needs to

write information to the aux file every time an entry is referenced. This needs to be read in on the second run and stored in a list corresponding to the appropriate glossary.

```
\newcommand*{\@print@noidx@glossary}{%
  \ifcsdef{@glsref@\@glo@type}%
  {%
```

Sort the entries:

```
\ifcsdef{@glo@sortmacro@\@glo@sorttype}%
  {%
  \csuse{@glo@sortmacro@\@glo@sorttype}{\@glo@type}%
  }%
  {%
  \PackageError{glossaries}{Unknown sort handler '\@glo@sorttype'}{}}%
  }%
```

Do the glossary heading and preamble

```
\glossarysection[\glossarytoctitle]{\glossarytitle}%
\glossarypreamble
```

The glossary style might use a tabular-like environment, which may cause scoping problems when setting the current letter group. The predefined tabular-like styles don't support letter group headings, but there's nothing to stop the user from defining their own custom style that might, so any redefinition of this command within theglossary will have to be done globally.

```
\def\@gls@currentlettergroup{%
\begin{theglossary}%
\glossaryheader
\glsresetentrylist
```

Iterate through the entries.

```
\forlistcsloop{\@gls@noidx@do}{@glsref@\@glo@type}%
```

Finally end the glossary and do the postamble:

```
\end{theglossary}%
\glossarypostamble
}%
{%
\@gls@noref@warn{\@glo@type}%
}%
}
```

`\glo@grabfirst` This won't work for UTF-8 and inputenc.

```
\def\glo@grabfirst#1#2\@nil{%
  \def\@gls@firsttok{#1}%
  \ifdefempty\@gls@firsttok
  {%
  \def\@glo@thislettergrp{0}%
  }%
  {%
```

Sanitize it:

```
\@onelevel@sanitize\@gls@firsttok
```

Fetch the first letter:

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

`\@glo@grabfirst`

```
\def\@glo@grabfirst#1#2\@nil{%  
  \ifdefempty\@glo@thislettergrp  
  {%  
    \def\@glo@thislettergrp{glssymbols}%  
  }%  
  {%  
    \count@=\uccode‘#1\relax  
    \ifnum\count@=0\relax  
    \def\@glo@thislettergrp{glssymbols}%  
  \else  
    \ifdefstring\@glo@sorttype{case}%  
    {%  
      \count@=‘#1\relax  
    }%  
    {%  
    }%  
  \edef\@glo@thislettergrp{\the\count@}%  
  \fi  
}%  
}
```

`\@gls@noidx@do` Handler for list iteration used by `\@print@noidx@glossary`. The argument is the entry label. This only allows one sublevel.

```
\newcommand{\@gls@noidx@do}[1]{%
```

Get this entry’s location list

```
\global\letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
```

Does this entry have a parent?

```
\ifglshasparent{#1}%  
{%
```

Has a parent.

```
\gls@level=\csuse{glo@\glsdetoklabel{#1}@level}\relax  
\ifdefvoid{\@gls@loclist}  
{%  
  \subglossentry{\gls@level}{#1}{}%  
}%  
{%  
  \subglossentry{\gls@level}{#1}%  
  {%  
    \glsnoidxprenumberlist{#1}%  
  }%  
}
```

```

        \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
    }%
} %
} %
{ %
Doesn't have a parent Get this entry's sort key
    \letcs{\@gls@sort}{glo\glsdetoklabel{#1}@sort}%
Fetch the first letter:
    \expandafter\glo@grabfirst\@gls@sort{}{}\@nil
    \ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
    {}%
    { %
Do the group header:
    \ifdefempty{\@gls@currentlettergroup}{}%
    { %
The group skip may start a new scope, so make a global assignment.
    \global\let\@glo@thislettergrp\@glo@thislettergrp
    \glsgroupskip
    }%
    \glsgroupheading{\@glo@thislettergrp}%
    }%

    \global\let\@gls@currentlettergroup\@glo@thislettergrp
Do this entry:
    \ifdefvoid{\@gls@loclist}
    { %
    \glossentry{#1}{}%
    }%
    { %
    \glossentry{#1}%
    { %
    \glsnoidxprenumberlist{#1}%
    \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
    }%
    }%
    }%
}
}

```

\glsnoidxloclist

\glsnoidxloclist{*(list cs)*}

Display location list.

```

\newcommand*{\glsnoidxloclist}[1]{%
  \def\@gls@noidxloclist@sep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxloclisthandler}{#1}%
}

```


`\glsnoidxloclisthandler` Handler for location list iterator.

```
\newcommand*\glsnoidxloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
```

Same as previous location so skip.

```
}%
  {%
    \@gls@noidxloclist@sep
    #1%
    \def\@gls@noidxloclist@sep{\delimN}%
    \def\@gls@noidxloclist@prev{#1}%
  }%
}
```

`\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 nonnumberlist key determines if this glossary should have a number list.

```
\define@boolkey{printgloss}[gls]{nonnumberlist}[true]{%
\ifglsnonnumberlist
  \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
}
```

```

        \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@glstetoklabel{#1}@name}%
    \expandafter\glsnamefont\expandafter{\glo@name}%
  }%
}
```

`\Glossentryname`

```
\newcommand*\Glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsnamefont{\Glsentryname{#1}}%
  }%
}
\glsmfuaddmap{\glossentryname}{\Glossentryname}
```

```
\glossentrydesc
  \newcommand*\glossentrydesc[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \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]{%
```

Note that `\dtl@ifsingle` (provided by `datatool`) is being phased out.

```
\def#2{#1}%
\regex_match:nnT { ^\A(.|glsnumbers|glssymbols)\Z } { #1 }
{ \ifcsundef{#1groupname}{ }\letcs#2{#1groupname} }
}
\ExplSyntaxOff
```

`\@gls@noidx@getgrouptitle` Version for the no-indexing app option:

```
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%
  \DTLifint{#1}%
  {\edef#2{\char#1\relax}}%
  {%
    \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
  }%
}
```

`\glsgetgrouplabel{<title>}`

This command does the reverse to the previous command. The argument is the group title, and it produces the group label. Note that if you redefine `\glsgetgrouptitle`, you will also need to redefine `\glsgetgrouplabel`.

`\glsgetgrouplabel` This only works for the symbol and number groups, so is best avoided and not documented. However, it has been converted to use \LaTeX 3 commands.

```
\ExplSyntaxOn
```

```

\newcommand*{\glsgetgrouplabel}[1]{%
\str_if_eq:eeTF {#1} { \glsymbolsgroupname }
{ glssymbols }
{
\str_if_eq:eeTF {#1} { \glsnumbersgroupname }
{ glnumbers }
{ #1 }
}
}
\ExplSyntaxOff

```

The command `\setentrycounter` sets the entry’s associated counter (required by `\glsnumber` 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][1]{%
\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.

```

\gls@wrglossdisableanchorcmds
\protected@edef\gls@tmp{\gls@entrycounter\gls@counterprefix
\gls@wrglosslocationtarget{\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\gls@wrglosslocationtextfmt{\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]{\abbrv}{\long}{\key-val list}

```

This emulates the way the old `glossary` package defined acronyms. It is equivalent to `\newacronym[\key-val list]{\label}{\abbrv}{\long}` and it additionally defines the command `\label` which is equivalent to `\gls{\label}` (thus `\label` must only contain alphabetical characters). If `\label` is omitted, `\abbrv` is used. This only emulates the syntax of the old `glossary` package. The way the acronyms appear in the list of acronyms is determined by the definition of `\newacronym` and the `glossary` style.

Note that `\label` can't have an optional argument if the `xspace` package is loaded. If `xspace` hasn't been loaded then you can do `\label[insert]` but

you can't do `\<label>[\<key-val list>]`. For example if you define the acronym `svm`, then you can do `\svm['s]` but you can't do `\svm[format=textbf]`. If the `xspace` package is loaded, `\svm['s]` will appear as `svm ['s]` which is unlikely to be the desired result. In this case, you will need to use `\gls` explicitly, e.g. `\gls{svm}['s]`. Note that it is up to the user to load `xspace` if desired.

```
\newcommand{\oldacronym}[4][\gls@label]{%
  \def\gls@label{#2}%
  \newacronym[#4]{#1}{#2}{#3}%
  \ifcsundef{xspace}%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}%
    }%
  }%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
        \noexpand\gls{#1}\noexpand\xspace}%
      }%
    }%
  }
}
```

```
\newacronym[\<key-val list>]{\<label>}{\<abbrev>}{\<long>}
```

This is a quick way of defining acronyms, using `\newglossaryentry` with the appropriate values. It sets the glossary type to `\acronymtype` which will be `acronym` if the package option `acronym` has been used, otherwise it will be the default glossary. Since `\newacronym` merely calls `\newglossaryentry`, the acronym is treated like any other glossary entry.

If you prefer a different format, you can redefine `\newacronym` as required. The optional argument can be used to override any of the settings.

This is just a stub. It's redefined by `\SetGenericNewAcronym` and by `glossaries-extra`. It used to also be redefined by the obsolete acronym styles that have now been removed. If it is used then it means that the acronym style hasn't been initialised.

```
\newacronym
\newcommand{\newacronym}[4][ ]{%
  \gls@newacronym{#1}{#2}{#3}{#4}%
}
```

```
\gls@newacronym
\newcommand{\gls@newacronym}[4]{%
```

Setting the acronym style will redefine `\newacronym` because it calls `\SetGenericNewAcronym` but if something weird has happened to alter this, redefine `\gls@newacronym` now to prevent an infinite loop.


```

\renewcommand{\gls@newacronym}[4]{%
\PackageError{glossaries}{incompatible acronym style:
rollback may be required}%
{something has gone wrong with \string\newacronym. Try
\string\usepackage{glossaries}[=v4.49]}%
}%
\setacronymstyle{long-short}%
\newacronym[#1]{#2}{#3}{#4}%
}

```

Set up some convenient short cuts. These need to be changed if `\newacronym` is changed (or if the description key is changed).

`\acrpluralsuffix` Plural suffix used by `\newacronym`. This just defaults to `\glspluralsuffix` but is changed to include `\textup` if the smallcaps option is used, so that the suffix doesn't appear in small caps as it doesn't look right. For example, ABCS looks as though the “s” is part of the acronym, but ABCs looks as though the “s” is a plural suffix. Since the entire text `abcs` is set in `\textsc`, `\textup` is need to cancel it out.

```
\newcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}
```

If `garamondx` has been loaded, need to use `\textulc` instead of `\textup`.

```

\glstextup
\newrobustcmd*{\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}

```

The following are defined for compatibility with version 2.07 and earlier.

```

\glsshortkey
\newcommand*{\glsshortkey}{short}

```

```

\glsshortpluralkey
\newcommand*{\glsshortpluralkey}{shortplural}

```

```

\glslongkey
\newcommand*{\glslongkey}{long}

```

```

\glslongpluralkey
\newcommand*{\glslongpluralkey}{longplural}

```

Using `\acrlinkfullformat` and `\acrfullformat` is now deprecated. However, these commands are still used in the initial definition of `\acrfull` and `\glsentryfull`. The acronym style now redefines `\glsentryfull`, and `\acrfull` now uses `\acrfullfmt`, which is redefined by the acronym style. Similar for the case and plural variants. This means that `\acrlinkfullformat` and `\acrfullformat` should now longer be used with v4.50, which will ensure that `\setacronymstyle` is used on the first instance of `\newacronym` if a style hasn't been set.

Outlier cases may occur where someone has defined an entry explicitly with `\newglossaryentry` and set the long and short keys and is using `\glsentryfull` or `\acrfull` without a style set. So, for now, the deprecated `\acrlinkfullformat` and `\acrfullformat` are retained even though the remaining obsolete acronym code has been removed. However, they may be removed in a future version.

`\acrlinkfullformat` Format for full links like `\acrfull`. Syntax: `\acrlinkfullformat{⟨long cs⟩}{⟨short cs⟩}{⟨options⟩}{⟨label⟩}{⟨insert⟩}` Deprecated.

```
\newcommand{\acrlinkfullformat}[5]{%
  \acrfullformat{#1{#3}{#4}[#5]}{#2{#3}{#4}[]}%
}
```

`\acrfullformat` Default full form is *⟨long⟩* (*⟨short⟩*). Deprecated.

```
\newcommand{\acrfullformat}[2]{#1\glsspace{#2}}
```

`\glsspace` Robust space to ensure it's written to the `.glsdefs` file.

```
\newrobustcmd{\glsspace}{\space}
```

The following `\acrfullfmt` set of commands are used by the corresponding `\acrfull` command, and are redefined by the acronym style to match the style's full form.

`\acrfullfmt` No case change full format. This default will be overridden by the acronym style, so this definition shouldn't actually be used any more, but it still needs to be defined so the it doesn't cause a problem when acronym styles redefine it. This is initialised to use the deprecated `\acrlinkfullformat` but will be redefined by the acronym style when it's set.

```
\newcommand*{\acrfullfmt}[3]{%
  \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
```

`\Acrfullfmt` Sentence case full format.

```
\newcommand*{\Acrfullfmt}[3]{%
  \acrlinkfullformat{\@Acrlong}{\@Acrshort}{#1}{#2}{#3}%
}
```

`\ACRfullfmt` All upper case full format.

```
\newcommand*{\ACRfullfmt}[3]{%
  \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
```

`\acrfullplfmt` No case change plural full format.

```
\newcommand*{\acrfullplfmt}[3]{%
  \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
```

```

\Acrfullplfmt Sentence case plural full format.
  \newcommand*\Acrfullplfmt}[3]{%
    \acrlinkfullformat{\@Acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
  }

\ACRfullplfmt All upper case plural full format.
  \newcommand*\ACRfullplfmt}[3]{%
    \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
  }

\acrfull Full form of the acronym.
  \newrobustcmd*\acrfull{\@gls@hyp@opt\ns@acrfull}

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

\@acrfull Low-level macro:
  \def\@acrfull#1#2[#3]{%
Make it easier for acronym styles to change this:
    \acrfullfmt{#1}{#2}{#3}%
  }

  Default format for full acronym

\Acrfull
  \newrobustcmd*\Acrfull{\@gls@hyp@opt\ns@Acrfull}
Add sentence case mapping.
  \glsmfuaddmap{\acrfull}{\Acrfull}

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

Low-level macro:
  \def\@Acrfull#1#2[#3]{%
Make it easier for acronym styles to change this:
    \Acrfullfmt{#1}{#2}{#3}%
  }

\ACRfull
  \newrobustcmd*\ACRfull{\@gls@hyp@opt\ns@ACRfull}

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

```

Low-level macro:

```
\def\@ACRfull#1#2[#3]{%
```

Make it easier for acronym styles to change this:

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

Plural:

```
\acrfullpl
```

```
\newrobustcmd*{\acrfullpl}{\@gls@hyp@opt\ns@acrfullpl}
```

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

Low-level macro:

```
\def\@acrfullpl#1#2[#3]{%
```

Make it easier for acronym styles to change this:

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

```
\Acrfullpl
```

```
\newrobustcmd*{\Acrfullpl}{\@gls@hyp@opt\ns@Acrfullpl}
```

Add sentence case mapping.

```
\glsmfuaddmap{\acrfullpl}{\Acrfullpl}  
  
\newcommand*\ns@Acrfullpl[2][]{%  
  \new@ifnextchar[{\@Acrfullpl{#1}{#2}}%  
    {\@Acrfullpl{#1}{#2}[ ]}%  
}
```

Low-level macro:

```
\def\@Acrfullpl#1#2[#3]{%
```

Make it easier for acronym styles to change this:

```
\Acrfullplfmt{#1}{#2}{#3}%  
}
```

```
\ACRfullpl
```

```
\newrobustcmd*{\ACRfullpl}{\@gls@hyp@opt\ns@ACRfullpl}
```

```
\newcommand*\ns@ACRfullpl[2][]{%  
  \new@ifnextchar[{\@ACRfullpl{#1}{#2}}%  
    {\@ACRfullpl{#1}{#2}[ ]}%  
}
```

Low-level macro:

```
\def\@ACRfullpl#1#2[#3]{%
```

Make it easier for acronym styles to change this:

```
\ACRfullplfmt{#1}{#2}{#3}%  
}
```

2.18 Predefined acronym styles

- `\acronymfont` Used by the acronym styles to apply a font change to the short form.
`\newcommand{\acronymfont}[1]{#1}`
- `\firstacronymfont` Used by the acronym styles to apply a font change to the short form on first use.
`\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}`
- `\acrnameformat` The styles that allow an additional description use `\acrnameformat{<short>}{<long>}` to determine what information is displayed in the name.
`\newcommand*{\acrnameformat}[2]{\acronymfont{#1}}`
- Define some tokens used by `\newacronym`:
- `\glskeylisttok`
`\newtoks\glskeylisttok`
- `\glslabeltok`
`\newtoks\glslabeltok`
- `\glsshorttok`
`\newtoks\glsshorttok`
- `\glslongtok`
`\newtoks\glslongtok`
- `\newacronymhook` Provide a hook for `\newacronym`:
`\newcommand*{\newacronymhook}{}`
- `\SetGenericNewAcronym` New improved version of setting the acronym style.
`\newcommand*{\SetGenericNewAcronym}{%`
- Change the behaviour of `\Glsentryname` to workaround expansion issues that cause a problem for `\makefirstuc`
`\let\@Gls@entryname\@Gls@acrentryname`
- Change the way acronyms are defined:
`\renewcommand{\newacronym}[4][]{%`
`\ifdefempty{\@glsacronymlists}{%`
`{%`
`\def\@glo@type{\acronymtype}{%`
`\setkeys{glossentry}{##1}{%`
`\DeclareAcronymList{\@glo@type}{%`
`}%`
`{}`
`\glskeylisttok{##1}{%`
`\glslabeltok{##2}{%`
`\glsshorttok{##3}{%`
`\glslongtok{##4}{%`

```

\newacronymhook
\protected@edef\do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\expandonce{\acronymentry{##2}}},%
    sort={\acronymsort{\the\glsshorttok}{\the\glslongtok}},%
    text={\the\glsshorttok},%
    short={\the\glsshorttok},%
    shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
    long={\the\glslongtok},%

    longplural={\the\glslongtok\noexpand\glspluralsuffix},%
    \GenericAcronymFields,%
    \the\glskeylisttok
  }%
}%
\do@newglossaryentry
}%

```

Make sure that `\acrfull` etc reflects the new style:

```

\renewcommand*\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}}%
\renewcommand*\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
\renewcommand*\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \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{\glsentryshort{#1}}
```

Default sort format for acronyms.

```
\newcommand*\acronymfont}[1]{#1}
```

\setacronymstyle

```
\setacronymfont{\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{\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` (*long*) (*short*) acronym style.

```
\newacronymstyle{long-short}%
{%
```

Check for long form in case this is a mixed glossary.

```
\ifglshaslong{\glslabel}{\glsacrfmt}{\glsacrfmt}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\genacrfullformat}[2]{%
    \glsentrylong{##1}##2\space
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*{\Genacrfullformat}[2]{%
    \Glsentrylong{##1}##2\space
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*{\genplacrfullformat}[2]{%
    \glsentrylongpl{##1}##2\space
    (\protect\firstacronymfont{\glsentryshortpl{##1}})%
  }%
  \renewcommand*{\Genplacrfullformat}[2]{%
    \Glsentrylongpl{##1}##2\space
    (\protect\firstacronymfont{\glsentryshortpl{##1}})%
  }%
  \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
  \renewcommand*{\acronymsort}[2]{##1}%
  \renewcommand*{\acronymfont}[1]{##1}%
  \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
```


`long-sp-short` Similar to the previous style but allows the space between the long and short form to be customized.

```

\newacronymstyle{long-sp-short}%
{%
Check for long form in case this is a mixed glossary.
\ifglshaslong{\glslabel}{\glsacfnacfmt}{\glsacnentryfmt}%
}%
{%
\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}{\glsacfnacfmt}{\glsacnentryfmt}%
}%
{%

```

```

\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshort{##1}}##2\space
  (\glsentrylong{##1})%
}%
\renewcommand*\Genacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshort{##1}}##2\space
  (\glsentrylong{##1})%
}%
\renewcommand*\genplacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
  (\glsentrylongpl{##1})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
  (\glsentrylongpl{##1})%
}%

\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}

```

long-sc-short *<long>* (`\textsc{<short>}`) acronym style.

```

\newacronymstyle{long-sc-short}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
}

```

long-sm-short *<long>* (`\textsmaller{<short>}`) acronym style.

```

\newacronymstyle{long-sm-short}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*\acrpluralsuffix{\glsacrpluralsuffix}%
}

```

sc-short-long *<short>* (`\textsc{<long>}`) acronym style.

```

\newacronymstyle{sc-short-long}%
{%

```

```

    \GlsUseAcrEntryDispStyle{short-long}%
  }%
  {%
    \GlsUseAcrStyleDefs{short-long}%
    \renewcommand{\acronymfont}[1]{\textsc{##1}}%
    \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
  }

```

`sm-short-long` $\langle short \rangle$ ($\backslash\text{textsmaller}\{\langle long \rangle\}$) acronym style.

```

\newacronymstyle{sm-short-long}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}

```

`long-short-desc` $\langle long \rangle$ ($\{\langle short \rangle\}$) acronym style that has an accompanying description (which the user needs to supply).

```

\newacronymstyle{long-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
  }

```

`long-sp-short-desc` $\langle long \rangle$ ($\{\langle short \rangle\}$) acronym style that has an accompanying description (which the user needs to supply). The space between the long and short form is given by $\backslash\text{glsacspace}$.

```

\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}%
{%
  \GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sc-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
```

`long-sm-short-desc` $\langle long \rangle$ (`\textsmaller{ $\langle short \rangle$ }`) acronym style that has an accompanying description (which the user needs to supply).

```
\newacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
```

`short-long-desc` $\langle short \rangle$ (`{ $\langle long \rangle$ }`) acronym style that has an accompanying description (which the user needs to supply).

```
\newacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##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}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
}
```

```

\GlsUseAcrStyleDefs{sc-short-long}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%
  \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}

```

sm-short-long-desc *<long>* (`\textsmaller{<short>}`) acronym style that has an accompanying description (which the user needs to supply).

```

\newacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%
  \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}

```

dua *<long>* only acronym style.

```

\newacronymstyle{dua}%
{%

```

Check for long form in case this is a mixed glossary.

```

\ifdefempty\glscustomtext
{%
  \ifglshaslong{\glslabel}%
  {%
    \glcifplural
  }%
}

```

Plural form:

```

\glscapscase
{%

```

Plural form, don't adjust case:

```

\glentrylongpl{\glslabel}\glinsert
}%
{%

```

Plural form, make first letter upper case:

```

\Glsentrylongpl{\glslabel}\glinsert
}%
{%

```

Plural form, all caps:

```

\glsupercase
{\glentrylongpl{\glslabel}\glinsert}%
}%

```

```

    }%
    {%
Singular form
    \glscapscase
    {%
Singular form, don't adjust case:
    \glstentrylong{\glslabel}\glsinsert
    }%
    {%
Subsequent singular form, make first letter upper case:
    \Glstentrylong{\glslabel}\glsinsert
    }%
    {%
Subsequent singular form, all caps:
    \glssupercase
    {\glstentrylong{\glslabel}\glsinsert}%
    }%
    }%
    }%
    {%
Not an acronym:
    \glsgenentryfmt
    }%
    }%
    {\glscustomtext\glsinsert}%
    }%
    {%
    \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%

    \renewcommand*{\acrfullfmt}[3]{%
    \glslink[##1]{##2}{\glstentrylong{##2}##3\space
    (\acronymfont{\glstentryshort{##2}})}%
    \renewcommand*{\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{\Glstentrylong{##2}##3\space
    (\acronymfont{\glstentryshort{##2}})}%
    \renewcommand*{\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
    \glssupercase{\glstentrylong{##2}##3\space
    (\acronymfont{\glstentryshort{##2}})}%
    }%
    }%

    \renewcommand*{\acrfullplfmt}[3]{%
    \glslink[##1]{##2}{\glstentrylongpl{##2}##3\space
    (\acronymfont{\glstentryshortpl{##2}})}%
    }%

    \renewcommand*{\Acrfullplfmt}[3]{%
    \glslink[##1]{##2}{\Glstentrylongpl{##2}##3\space
    (\acronymfont{\glstentryshortpl{##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}{\glsacfm}{\glsacentryfmt}%
}%
{%
  \renewcommand*\GenericAcronymFields}{description={\the\glslongtok}}%
}

```

Need to ensure hyperlinks are switched off on first use:

```

\glshyperfirstfalse
\renewcommand*\genacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshort{##1}}##2%
  \protect\footnote{\glsentrylong{##1}}%
}%

```

```

\renewcommand*{\Genacrfullformat}[2]{%
  \firstacronymfont{\Glsentryshort{##1}}##2%
  \protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%

```

Don't use footnotes for \acrfull:

```

\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
  \glsuppercase{\acronymfont{\glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
  \glsuppercase{\acronymfont{\glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}}}%

```

Similarly for \glsentryfull etc:

```

\renewcommand*{\glsentryfull}[1]{%
  \acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\glsentryfullpl}[1]{%
  \acronymfont{\glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
}

```


footnote-sc \textsc{<short>}\footnote{<long>} acronym style.

```
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}%
```

footnote-sm \textsmaller{<short>}\footnote{<long>} acronym style.

```
\newacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}%
```

footnote-desc <short>\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
\newacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
  }
}
```

footnote-sc-desc \textsc{<short>}\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
\newacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%

```

```

        \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

```

\Ac1
    \let\Ac1\Acrlong
    First letter upper case plural long form
\Ac1p
    \let\Ac1p\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]{%
  \defglsentryfmt[#1]{\glsentryfmt}%
}
```

The following commands for setting the acronym style are now deprecated.

`\DefaultNewAcronymDef` Sets up the acronym definition for the default style. The information is provided by the tokens `\glslabeltok`, `\glsshorttok`, `\glslongtok` and `\glskeylisttok`. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\noexpand@assign@default@plurals` `\@gls@noexpand@assign@default@plurals` was used in `\DefaultNewAcronymDef`. Added in v4.47 to fix bug #162 (<https://www.dickimaw-books.com/bugtracker.php?key=162>). Removed in v4.50 along with the deprecated styles.

`\@gls@default@acr@plural`

`\gls@default@acr@firstplural`

`\gls@default@acr@descplural`

`\SetDefaultAcronymStyle` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\acrfootnote` This was used by the old footnote acronym styles. Since it may have been used outside of those styles, it has been retained.

```
\newcommand*\acrfootnote[3]{\acrlinkfootnote{#1}{#2}{#3}}
```

`\acrlinkfootnote` This was used by the old footnote acronym styles. Since it may have been used outside of those styles, it has been retained.

```
\newcommand*\acrlinkfootnote[3]{%
  \footnote{\glslink[#1]{#2}{#3}}%
}
```

`\acrnofootnote` This was used by the old footnote acronym styles. Since it may have been used outside of those styles, it has been retained.

```
\newcommand*\acrnofootnote[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.

`\FootnoteAcronymDef` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`descriptionFootnoteAcronymStyle` If a description and footnote are both required, store the long form in the `symbol` key. Store the short form in `text` key. Note that since the long form is stored in the `symbol` key, if you want the long form to appear in the list of acronyms, you need to use a glossary style that displays the `symbol` key. This style was deprecated with the introduction of `\newacronymstyle`. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`descriptionDUAAcronymDisplayStyle` Sets the acronym display style for given glossary with description and dua combination. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`DescriptionDUANewAcronymDef` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`descriptionDUAAcronymStyle` Description, don't use acronym and no footnote. Note that the short form is stored in the `symbol` key, so if the short form needs to be displayed in the glossary, use a style that displays the `symbol`. This style was deprecated with the introduction of `\newacronymstyle`. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`descriptionAcronymDisplayStyle` Sets the acronym display style for given glossary using the description setting (but not footnote or dua). Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\DescriptionNewAcronymDef` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\SetDescriptionAcronymStyle` Option `description` is used, but not `dua` or `footnote`. Store long form in first key and short form in `text` and `symbol` key. The name is stored using `\acnameformat` to allow the user to override the way the name is displayed in the list of acronyms. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`FootnoteAcronymDisplayStyle` Sets the acronym display style for given glossary with footnote setting (but not description or dua). Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\FootnoteNewAcronymDef` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\SetFootnoteAcronymStyle` If footnote package option is specified, set the first use to append the long form (stored in `description`) as a footnote. Use the `description` key to store the long form. This style was deprecated with the introduction of `\newacronymstyle`. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\glsdoparenifnotempty` Do a space followed by the argument if the argument doesn't expand to empty or `\relax`. If argument isn't empty (or `\relax`), apply the macro to it given in the second argument.

```

\newrobustcmd*{\glsdoparenifnotempty}[2]{%
  \protected@edef\gls@tmp{#1}%
  \ifdefempty\gls@tmp
  {}%
  {%
    \ifx\gls@tmp\@gls@default@value
    \else
    \space (#2{#1})%
    \fi
  }%
}

```

`\SetSmallAcronymDisplayStyle` Sets the acronym display style for given glossary where neither footnote nor description is required, but smallcaps or smaller specified. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\SmallNewAcronymDef` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\SetSmallAcronymStyle` Neither footnote nor description required, but smallcaps or smaller specified. Use the `symbol` key to store the short form and `first` to store the long form. This style was deprecated with the introduction of `\newacronymstyle`. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\SetDUADisplayStyle` Sets the acronym display style for given glossary with dua setting. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\DUANewAcronymDef` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\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
}

\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@\glsdetoklabel{#1}@symbolplural\endcsname  
}
```

`\showgloshort`

```
\showgloshort{<label>}
```

```
\newcommand*{\showgloshort}[1]{%  
  \expandafter\show\csname glo@\glsdetoklabel{#1}@short\endcsname  
}
```

`\showglolong`

```
\showglolong{<label>}
```

```
\newcommand*{\showglolong}[1]{%  
  \expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname  
}
```

`\showgloindex`

```
\showgloindex{<label>}
```

```
\newcommand*{\showgloindex}[1]{%  
  \expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname  
}
```

`\showgloflag`

```
\showgloflag{<label>}
```

```
\newcommand*{\showgloflag}[1]{%  
  \expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname  
}
```

`\showgloloclist`

```
\showgloloclist{<label>}
```

```
\newcommand*{\showgloloclist}[1]{%  
  \expandafter\show\csname glo@\glsdetoklabel{#1}@loclist\endcsname  
}
```

`\showglofield`

```
\showglofield{<label>}{<field>}
```

```
\newcommand*{\showglofield}[2]{%
  \csshow{glo@glstoklabel{#1}@#2}%
}
```

`\showacronymlists`

```
\showacronymlists
```

Show list of glossaries that have been flagged as a list of acronyms.

```
\newcommand*{\showacronymlists}{%
  \show@glstacronymlists
}
```

`\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 gloolist@#1\endcsname
}
```

2.21 Compatibility with version 2.07 and below

The backward-compatibility options have been removed in version 4.50. Use the rollback feature instead if backward-compatibility is required. Bear in mind that v2.07 was released 2010-07-10 and v3.07 was released 2013-07-05. Any old documents from that long ago will probably have problems with recent kernels.

In order to fix some bugs in v3.0, it was necessary to change the way information is written to the `glo` file, which also meant a change in the format of the Xindy style file. The compatibility option is meant for documents that use a customised Xindy style file with `\noist`. With the compatibility option, hopefully `xindy` will still be able to process the old document, but the bugs will remain. The issues in versions 2.07 and below:

- With `xindy`, the counter used by the entry was hard-coded into the Xindy style file. This meant that you couldn't use the `counter` to swap counters.
- With both `xindy` and `makeindex`, if used with `hyperref` and `\theH<counter>` was different to `\thecounter`, the link in the location number would be undefined.

3 Glossary Styles

3.1 Glossary hyper-navigation definitions (`glossary-hypernav.sty`)

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-hypernav-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-hypernav-2021-11-01.sty}
\DeclareRelease{v4.52}{2022-11-03}{glossary-hypernav-2022-11-03.sty}
\DeclareCurrentRelease{v4.53}{2023-09-29}
```

Package Definition:

```
\ProvidesPackage{glossary-hypernav}[2023/09/29 v4.53 (NLCT)]
```

The commands defined in this package are provided to help navigate around the groups within a glossary (see [subsection 2.16](#).) `\printglossary` (and `\printglossaries`) set `\@glo@type` to the label of the current glossary. This is used to create a unique hypertarget in the event of multiple glossaries.

```
\glsnavhyperlink[<type>]{<label>}{<text>}
```

This command makes `<text>` a hyperlink to the glossary group whose label is given by `<label>` for the glossary given by `<type>`.

`\glsnavhyperlink` The optional argument is the glossary type and can be used to make the group label unique in case the same group occurs in multiple glossaries. The optional argument was `\@glo@type` but is now `\currentglossary` because that's less likely to change within the glossary. It's unlikely that the optional argument will be used to set it to anything other than `\currentglossary` but it makes it possible to hyperlink to the group from outside of the glossary.

```
\newcommand*{\glsnavhyperlink}[3][\currentglossary]{%  
  
  \protected@edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%  
  \@glslink{\glsnavhyperlinkname{#1}{#2}}{#3}%  
}
```

`\glsnavhyperlinkname` Expands to the hypertarget name. The first argument is the glossary type. The second argument is the group label.

```
\newcommand*{\glsnavhyperlinkname}[2]{glsn:#1@#2}
```

Version 4.53: reimplemented list of hypertarget groups as a sequence. Previously, the targets were stored as comma-separated lists in `\@gls@hypergrouplist@<glossary>` (where `<glossary>` is the glossary label). They are now stored in a sequence `\g_glossaries_<glossary>_hypergroups_seq`.

```
\ExplSyntaxOn
```

```
\glossaries_add_hypergroup:nn{<glossary>}{<target>}
```

Add a target to the sequence unless it's already contained in it:

```
\cs_new:Npn \glossaries_add_hypergroup:nn #1 #2  
{  
  \seq_if_exist:cTF  
  { g_glossaries_ #1 _hypergroups_seq }  
  {  
    \seq_if_in:cnF  
    { g_glossaries_ #1 _hypergroups_seq } { #2 }  
    {  
      \seq_gput_right:cn  
      { g_glossaries_ #1 _hypergroups_seq }  
      { #2 }  
    }  
  }  
}
```

```

    }
  }
  {
    \seq_new:c { g_glossaries_ #1 _hypergroups_seq }
    \seq_gput_right:cn
      { g_glossaries_ #1 _hypergroups_seq }
      { #2 }
  }
}
\cs_generate_variant:Nn \glossaries_add_hypergroup:nn { nx }

```

Check if the given target has been added to the sequence to avoid duplicates.

```

\cs_new:Npn \glossaries_if_hypergroup:nnTF #1 #2 #3 #4
{
  \seq_if_exist:cTF
    { g_glossaries_ #1 _hypergroups_seq }
    {
      \seq_if_in:cnTF
        { g_glossaries_ #1 _hypergroups_seq } { #2 }
        { #3 } { #4 }
    }
  { #4 }
}
\cs_generate_variant:Nn \glossaries_if_hypergroup:nnTF { nxTF }

```

Only true part:

```

\cs_new:Npn \glossaries_if_hypergroup:nnT #1 #2 #3
{
  \glossaries_if_hypergroup:nnTF { #1 } { #2 } { #3 } { }
}
\cs_generate_variant:Nn \glossaries_if_hypergroup:nnT { nxT }

```

Only false part:

```

\cs_new:Npn \glossaries_if_hypergroup:nnF #1 #2 #3
{
  \glossaries_if_hypergroup:nnTF { #1 } { #2 } { } { #3 }
}
\cs_generate_variant:Nn \glossaries_if_hypergroup:nnF { nxF }

```

```
\glsnavhypertarget[type]{label}{text}
```

`\glsnavhypertarget`

This command makes *text* a hypertext for the glossary group whose label is given by *label* in the glossary given by *type*. If *type* is omitted, `\glo@type` is used which is set by `\printglossary` to the current glossary label. The optional argument complicates things for `glossaries-extra`.

```
\newcommand*{\glsnavhypertarget}{\protect\@gls@navhypertarget}
```

`\@gls@navhypertarget`

```
\newcommand*{\@gls@navhypertarget}[3][\currentglossary]{%
```

```

    \@glsnavhypertarget{#1}{#2}{#3}%
  }

```

`\glsnavhypergroupdotarget` Provided to make it easier for glossaries-extra to adjust the target.

```

\newcommand{\glsnavhypergroupdotarget}[3]{
  \@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.

```

  \glsnavhypergroupdotarget { #1 } { #2 } { #3 }

```

Check list of known groups to determine if a re-run is required.

```

  \glossaries_if_hypergroup:nxF { #1 } { #2 }
  {

```

This group was not included in the list, so issue a warning.

```

    \GlossariesWarningNoLine{Navigation ~ panel ~
      for ~ glossary ~ type ~ '#1'^^Jmissing ~ group ~ '#2'}%

```

```

  \gdef\gls@hypergrouprerun{%
    \GlossariesWarningNoLine{Navigation ~ panel ~
      has ~ changed. ~ Rerun ~ LaTeX}}%
  }

```

```

}

```

`\gls@hypergrouprerun` Give a warning at the end if re-run required

```

\let\gls@hypergrouprerun\relax
\AtEndDocument{\gls@hypergrouprerun}

```

`\@gls@hypergroup` This adds to (or creates) the command `\@gls@hypergrouplist@<glossary type>` which lists all groups for a given glossary, so that the navigation bar only contains those groups that are present. However it requires at least 2 runs to ensure the information is up-to-date.

```

\newcommand*{\@gls@hypergroup}[2]{
  \glossaries_add_hypergroup:nx { #1 } { #2 }
}

```

The `\glsnavigation` command displays a simple glossary group navigation. The symbol and number elements are defined separately, so that they can be suppressed if need be. (In earlier versions this command will produce a link to all 28 groups, but some groups may not be defined if there are groups that do not contain any terms, in which case you will get an undefined hyperlink warning. Version 1.14 changed this to only use labels for groups that are present.) Now for the whole navigation bit:


```

\glsnavigation
\newcommand*\glsnavigation}{%
\def\@gls@between{}%
\seq_map_inline:cn { g_glossaries_ \currentglossary _hypergroups_seq }
{
\@gls@between
\glsnavigationitem { ##1 }
\let\@gls@between\glshypernavsep
}%
}

```

```

\glsnavigationitem
\newcommand{\glsnavigationitem}[1]{%
\@gls@getgrouptitle { #1 } { \@gls@grptitle }
\exp_args:Nno \glsnavhyperlink { #1 } { \@gls@grptitle }
}

```

```

\glshypernavsep Separator for the hyper navigation bar.
\newcommand*\glshypernavsep}{\space\textbar\space}

```

The `\glssymbolnav` produces a simple navigation set of links for just the symbol and number groups. This used to be used at the start of `\glsnavigation`. This command is no longer needed.

```

\glssymbolnav
\newcommand*\glssymbolnav}{%
\glsnavhyperlink{glsymbols}{\glsgetgrouptitle{glsymbols}}%
\glshypernavsep
\glsnavhyperlink{glsnumbers}{\glsgetgrouptitle{glsnumbers}}%
\glshypernavsep
}
\ExplSyntaxOff

```

3.2 In-line Style (glossary-inline.sty)

This defines an in-line style where the entries are comma-separated with just the name and description displayed.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```

\DeclareRelease{v4.46}{2020-03-19}{glossary-inline-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-inline-2021-11-01.sty}
\DeclareCurrentRelease{v4.53}{2023-09-29}

```

Declare package:

```
\ProvidesPackage{glossary-inline}[2023/09/29 v4.53 (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}}%
  \glsentryitem{##2}%
  \glsinlinesubdescformat{\glossentrydesc{##2}}{\glossentrysymbol{##2}}{##3}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%
```

Nothing special between groups:

```
\renewcommand*{\glsgroupskip}{}%
}
```

`\glsinlineifhaschildren`

```
\newcommand{\glsinlineifhaschildren}[3]{%
  \ifglschildren{#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{v4.53}{2023-09-29}
```

Declare package:

```
\ProvidesPackage{glossary-list}[2023/09/29 v4.53 (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{\glsgetgrouptitle{##1}}]}
```

altlisthypergroup The altlisthypergroup glossary style is like the altlistgroup style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{altlisthypergroup}{%
```

Base it on the altlist style:

```
\setglossarystyle{altlist}%
```

Add navigation links at the start of the environment.

```
\renewcommand*{\glossaryheader}{%
  \glslistnavigationitem{\glsnavigation}}%
```

Each group has a heading with a hypertext:

```
\renewcommand*{\glsgroupheading}[1]{%
  \item[\glslistgroupheaderfmt
    {\glsnavhypertext{##1}{\glsgetgrouptitle{##1}}}]}
```

listdotted The listdotted glossary style was supplied by Axel Menzel. I've modified it slightly so that the distance from the start of the name to the end of the dotted line is specified by `\glslistdottedwidth`. Note that this style ignores the page numbers as well as the symbol. Sub-entries are displayed in the same way as top-level entries.

```
\newglossarystyle{listdotted}{%
```

Base it on the list style:

```
\setglossarystyle{list}%
```

Each main (level 0) entry starts a new item:

```
\renewcommand*{\glossentry}[2]{%
  \item[]\makebox[\glslistdottedwidth][l]{%
    \glsentryitem{##1}%
    \glstarget{##1}{\glossentryname{##1}}%
    \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##1}}%
```

Sub entries have the same format as main entries:

```
\renewcommand*{\subglossentry}[3]{%
  \item[]\makebox[\glslistdottedwidth][l]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\glossentryname{##2}}%
    \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##2}}%
}
```

`\glslistdottedwidth`

```
\newlength\glslistdottedwidth
\setlength{\glslistdottedwidth}{.5\hspace}
```

`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[\glstryitem{##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{v4.53}{2023-09-29}
```

Declare package:

```
\ProvidesPackage{glossary-long}[2023/09/29 v4.53 (NLCT)]
```

Requires the `longtable` package:

```
\RequirePackage{longtable}
```

`\glsdescwidth` This is a length that governs the width of the description column. (There's a chance that the user may specify `nolong` and then load `glossary-long` later, in which case `\glsdescwidth` may have already been defined by `glossary-super`. The same goes for `\glspagelistwidth`.)

```
\@ifundefined{glsdescwidth}{%
  \newlength\glsdescwidth
  \setlength{\glsdescwidth}{0.6\hsize}
}{}
```

`\glspagelistwidth` This is a length that governs the width of the page list column.

```
\@ifundefined{glspagelistwidth}{%
  \newlength\glspagelistwidth
  \setlength{\glspagelistwidth}{0.1\hsize}
}{}
```

`long` The long glossary style command which uses the `longtable` environment:

```
\newglossarystyle{long}{%
```

Use `longtable` with two columns:

```
\renewenvironment{theglossary}%
  {\begin{longtable}[lp{\glsdescwidth}]}%
  {\end{longtable}}%
```


Do nothing at the start of the environment:

```
\renewcommand*\glossaryheader}{}
```

No heading between groups:

```
\renewcommand*\glsgroupheading}[1]{}
```

Main (level 0) entries displayed in a row:

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
}%
```

Sub entries displayed on the following row without the name:

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
  ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip}{}%
\else
  \renewcommand*\glsgroupskip}{ & \tabularnewline}%
\fi
}
```

`longborder` The `longborder` style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{longborder}{%
```

Base it on the `glostylelong` style:

```
\setglossarystyle{long}%
```

Use `longtable` with two columns with vertical lines between each column:

```
\renewenvironment{theglossary}{%
  \begin{longtable}{|l|p{\glsdescwidth}|}{\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
\renewcommand*\glossaryheader}{\hline\endhead\hline\endfoot}%
}
```

`longheader` The `longheader` style is like the `long` style but with a header:

```
\newglossarystyle{longheader}{%
```

Base it on the `glostylelong` style:

```
\setglossarystyle{long}%
```

Set the table's header:

```
\renewcommand*\glossaryheader}{%
  \bfseries \entryname & \bfseries \descriptionname\tabularnewline\endhead}%
}
```

`longheaderborder` The `longheaderborder` style is like the `long` style but with a header and border:

```
\newglossarystyle{longheaderborder}{%
```

Base it on the `glostylelongborder` style:

```
\setglossarystyle{longborder}%
```

Set the table's header and add horizontal line to table's foot:

```
\renewcommand*\glossaryheader{%
  \hline\bfseries \entryname & \bfseries
  \descriptionname\tabularnewline\hline
  \endhead
  \hline\endfoot}%
}
```

`long3col` The `long3col` style is like `long` but with 3 columns

```
\newglossarystyle{long3col}{%
```

Use a `longtable` with 3 columns:

```
\renewenvironment{theglossary}%
  {\begin{longtable}[lp{\glsdescwidth}p{\glspagelistwidth}}%
  {\end{longtable}}%
```

No table header:

```
\renewcommand*\glossaryheader{}%
```

No headings between groups:

```
\renewcommand*\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
\renewcommand{\glossentry}[2]{%
  \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>)

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip}{ & & \tabularnewline}%
\fi
}
```

`long3colborder` The `long3colborder` style is like the `long3col` style but with a border:

```
\newglossarystyle{long3colborder}{%
Base it on the glostylelong3col style:
\setglossarystyle{long3col}%
Use a longtable with 3 columns with vertical lines around them:
\renewenvironment{theglossary}%
{\begin{longtable}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
{\end{longtable}}%
Place horizontal lines at the head and foot of the table:
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
```

`long3colheader` The `long3colheader` style is like `long3col` but with a header row:

```
\newglossarystyle{long3colheader}{%
Base it on the glostylelong3col style:
\setglossarystyle{long3col}%
Set the table's header:
\renewcommand*{\glossaryheader}{%
\bfseries\entryname&\bfseries\descriptionname&
\bfseries\pagelistname\tabularnewline\endhead}%
}
```

`long3colheaderborder` The `long3colheaderborder` style is like the above but with a border

```
\newglossarystyle{long3colheaderborder}{%
Base it on the glostylelong3colborder style:
\setglossarystyle{long3colborder}%
Set the table's header and add horizontal line at table's foot:
\renewcommand*{\glossaryheader}{%
\hline
\bfseries\entryname&\bfseries\descriptionname&
\bfseries\pagelistname\tabularnewline\hline\endhead
\hline\endfoot}%
}
```

`long4col` The `long4col` style has four columns where the third column contains the value of the associated symbol key.

```
\newglossarystyle{long4col}{%
Use a longtable with 4 columns:
\renewenvironment{theglossary}%
{\begin{longtable}{llll}}%
{\end{longtable}}%
No table header:
\renewcommand*{\glossaryheader}{}%
}
```

No group headings:

```
\renewcommand*\glsgroupheading}[1]{}
```

Main (level 0) entries on a single row (name in first column, description in second column, symbol in third column, page list in last column):

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  \glossentrysymbol{##1} &
  ##2\tabularnewline
}%
```

Sub entries on a single row with no name (description in second column, symbol in third column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glsesubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  \glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip{ & & \tabularnewline}%
\fi
}
```

`long4colheader` The `long4colheader` style is like `long4col` but with a header row.

```
\newglossarystyle{long4colheader}{%
```

Base it on the `glostylelong4col` style:

```
\setglossarystyle{long4col}%
```

Table has a header:

```
\renewcommand*\glossaryheader{%
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\endhead}%
}
```

`long4colborder` The `long4colborder` style is like `long4col` but with a border.

```
\newglossarystyle{long4colborder}{%
```

Base it on the `glostylelong4col` style:

```
\setglossarystyle{long4col}%
```

Use a `longtable` with 4 columns surrounded by vertical lines:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{|l|l|l|l|}}%
  {\end{longtable}}%
```

Add horizontal lines to the head and foot of the table:

```
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
```

`long4colheaderborder` The `long4colheaderborder` style is like the above but with a border.

```
\newglossarystyle{long4colheaderborder}{%
```

Base it on the `glostylelong4col` style:

```
\setglossarystyle{long4col}%
```

Use a `longtable` with 4 columns surrounded by vertical lines:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{|l|l|l|l|}}%
  {\end{longtable}}%
```

Add table header and horizontal line at the table's foot:

```
\renewcommand*{\glossaryheader}{%
  \hline\bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\hline\endhead
  \hline\endfoot}%
}
```

`altlong4col` The `altlong4col` style is like the `long4col` style but can have multiline descriptions and page lists.

```
\newglossarystyle{altlong4col}{%
```

Base it on the `glostylelong4col` style:

```
\setglossarystyle{long4col}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}%
  {\end{longtable}}%
}
```

`altlong4colheader` The `altlong4colheader` style is like `altlong4col` but with a header row.

```
\newglossarystyle{altlong4colheader}{%
```

Base it on the `glostylelong4colheader` style:

```
\setglossarystyle{long4colheader}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}%
  {\end{longtable}}%
}
```

`altlong4colborder` The `altlong4colborder` style is like `altlong4col` but with a border.

```
\newglossarystyle{altlong4colborder}{%
```

Base it on the `glostylelong4colborder` style:

```
\setglossarystyle{long4colborder}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{|l|p{\glsdescwidth}|l|p{\glspagelistwidth}|}%  
  {\end{longtable}}%  
}
```

`altlong4colheaderborder` The `altlong4colheaderborder` style is like the above but with a header as well as a border.

```
\newglossarystyle{altlong4colheaderborder}{%
```

Base it on the `glostylelong4colheaderborder` style:

```
\setglossarystyle{long4colheaderborder}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{|l|p{\glsdescwidth}|l|p{\glspagelistwidth}|}%  
  {\end{longtable}}%  
}
```

3.5 Glossary Styles using `longtable` and `booktabs` (`glossary-longbooktabs.sty`)

The styles here are based on David Carlisle's patch at <http://tex.stackexchange.com/a/56890>

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-longbooktabs-2020-03-19.sty}  
\DeclareRelease{v4.49}{2021-11-01}{glossary-longbooktabs-2021-11-01.sty}  
\DeclareCurrentRelease{v4.53}{2023-09-29}
```

Declare package:

```
\ProvidesPackage{glossary-longbooktabs}[2023/09/29 v4.53 (NLCT)]
```

Requires `booktabs` package:

```
\RequirePackage{booktabs}
```

and the base packages for long styles:

```
\RequirePackage{glossary-long}  
\RequirePackage{glossary-longragged}
```

(`longtable` and `array` loaded by those packages).

`long-booktabs` The `long-booktabs` style is similar to the `longheader` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{long-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

As with the `longheader` style, use the `long` style as a base.

```
\setglossarystyle{long}%
```

Add a header with rules.

```
\renewcommand*{\glossaryheader}{%
\toprule \bfseries \entryname & \bfseries
\descriptionname\tabularnewline\midrule\endhead
\bottomrule\endfoot}%
```

Check for the `nogroupskip` package option. If there should be a gap between groups, insert the penalty and the vertical space. The check for `nogroupskip` should occur outside `\glsgroupskip` to be on the safe side.

```
\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}
```

`long3col-booktabs` The `long3col-booktabs` style is similar to the `long3colheader` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{long3col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the `long3col` style as a base.

```
\setglossarystyle{long3col}%
```

Add a header with rules.

```
\renewcommand*{\glossaryheader}{%
\toprule \bfseries \entryname &
\bfseries \descriptionname &
\bfseries \pagelistname
\tabularnewline\midrule\endhead
\bottomrule\endfoot}%
```

Check for the `nogroupskip` package option. If there should be a gap between groups, insert the penalty and the vertical space. The check for `nogroupskip` should occur outside `\glsgroupskip` to be on the safe side.

```
\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
```

```

\else
  \renewcommand*\glsgroupskip{\glspenaltygroupskip}%
\fi
}

```

long4col-booktabs The `long4col-booktabs` style is similar to the `long4colheader` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{long4col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the `long4col` style as a base.

```
\setglossarystyle{long4col}{%
```

Add a header with rules.

```

\renewcommand*\glossaryheader}{%
  \toprule \bfseries \entryname &
  \bfseries \descriptionname &
  \bfseries \symbolname &
  \bfseries \pagelistname
  \tabularnewline\midrule\endhead
  \bottomrule\endfoot}%

```

Check for the `nogroupskip` package option. If there should be a gap between groups, insert the penalty and the vertical space. The check for `nogroupskip` should occur outside `\glsgroupskip` to be on the safe side.

```

\ifglsnogroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip{\glspenaltygroupskip}%
\fi
}

```

altlong4col-booktabs The `altlong4col-booktabs` style is similar to the `altlong4colheader` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{altlong4col-booktabs}{%
```

The patch `\glspatchLToutput` is already applied in `long4col-booktabs` and so doesn't need to be here.

```
\glspatchLToutput
```

Use the `long4col-booktabs` style as a base.

```
\setglossarystyle{long4col-booktabs}{%
```

Change the column specifications:

```

\renewenvironment{theglossary}{%
  {\begin{longtable}{lp{\glstdescwidth}lp{\glspagelistwidth}}}%
  {\end{longtable}}}%
}

```


Ragged styles.

longragged-booktabs The `longragged-booktabs` style is similar to the `longragged` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{longragged-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the `long-booktabs` style as a base.

```
\setglossarystyle{long-booktabs}{%
```

Adjust the column specification.

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{1>{\raggedright}p{\glsdescwidth}}}%  
  {\end{longtable}}%  
}
```

longragged3col-booktabs The `longragged3col-booktabs` style is similar to the `longragged3col` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{longragged3col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the `long3col-booktabs` style as a base.

```
\setglossarystyle{long3col-booktabs}{%
```

Adjust the column specification.

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{1>{\raggedright}p{\glsdescwidth}%  
    >{\raggedright}p{\glspagelistwidth}}}%  
  {\end{longtable}}%  
}
```

altlongragged4col-booktabs The `altlongragged4col-booktabs` style is similar to the `altlongragged4col` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{altlongragged4col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the `altlong4col-booktabs` style as a base.

```
\setglossarystyle{altlong4col-booktabs}{%
```

Adjust the column specification.

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{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\vsizel\@colroom
\copy\LT@head
\glsLTpenaltycheck
\nobreak
\fi
}%
}

```

3.6 Glossary Styles using longtable (glossary-longragged.sty)

The glossary styles defined in the `glossary-longragged` package used the `longtable` environment in the glossary and use ragged right formatting for the multiline columns.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```

\DeclareRelease{v4.46}{2020-03-19}{glossary-longragged-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-longragged-2021-11-01.sty}
\DeclareCurrentRelease{v4.53}{2023-09-29}

```

Declare package:

```
\ProvidesPackage{glossary-longragged}[2023/09/29 v4.53 (NLCT)]
```

Requires the `array` package:

```
\RequirePackage{array}
```

Requires the `longtable` package:

```
\RequirePackage{longtable}
```

`\glsdescwidth` This is a length that governs the width of the description column. This may have already been defined.

```

\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{}

```

`\glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined.

```

\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{}

```

`longragged` The `longragged` glossary style is like the `long` but uses ragged right formatting for the description column.

```
\newglossarystyle{longragged}{%
```

Use `longtable` with two columns:

```

\renewenvironment{theglossary}%
  {\begin{longtable}{1>{\raggedright}p{glsdescwidth}}}%
  {\end{longtable}}%

```

Do nothing at the start of the environment:

```
\renewcommand*\glossaryheader}{}
```

No heading between groups:

```
\renewcommand*\glsgroupheading}[1]{}
```

Main (level 0) entries displayed in a row:

```
\renewcommand{\glossentry}[2]{%
  \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 `glostyelongraggedborder` 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}[1>{\raggedright}p{\glsdescwidth}1%
   >{\raggedright}p{\glspagelistwidth}}}%
  {\end{longtable}}%
```

Table has a header:

```
\renewcommand*{\glossaryheader}{%
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\endhead}%
}
```

`altlongragged4colborder` The `altlongragged4colborder` style is like `altlongragged4col` but with a border.

```
\newglossarystyle{altlongragged4colborder}{%
```

Base it on the `glostylealtlongragged4col` style:

```
\setglossarystyle{altlongragged4col}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{|1|>{\raggedright}p{\glsdescwidth}|1|%
   >{\raggedright}p{\glspagelistwidth}|}}%
  {\end{longtable}}%
```

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}{|1|>{\raggedright}p{\glsdescwidth}|1|%
   >{\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{v4.53}{2023-09-29}
```

Declare package:

```
\ProvidesPackage{glossary-mcols}[2023/09/29 v4.53 (NLCT)]
```

Required packages:

```
\RequirePackage{multicol}
```

```
\RequirePackage{glossary-tree}
```

`\indexspace` There are a few classes that don't define `\indexspace`, so provide a definition if it hasn't been defined.

```
\providecommand{\indexspace}{%  
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax  
}
```

`\glsmcols` Define macro in which to store the number of columns. (Defaults to 2.)

```
\newcommand*{\glsmcols}{2}
```

`mcolindex` Multi-column index style. Same as the `index`, but puts the glossary in multiple columns. (Ideally the glossary title should go in the optional argument of `multicols`, but the title isn't part of the glossary style.)

```
\newglossarystyle{mcolindex}{%  
  \setglossarystyle{index}%  
  \renewenvironment{theglossary}%  
  {%  
    \begin{multicols}{\glsmcols}  
    \setlength{\parindent}{0pt}%  
    \setlength{\parskip}{0pt plus 0.3pt}%  
  
    \let\item\glstreeitem  
    \let\subitem\glstreesubitem  
    \let\subsubitem\glstreesubsubitem  
  }%  
  {\end{multicols}}%  
}
```

`mcolindexgroup` As `mcolindex` but has headings:

```
\newglossarystyle{mcolindexgroup}{%  
  \setglossarystyle{mcolindex}%  
  \renewcommand*{\glsgroupheading}[1]{%  
    \item\glstreegroupheaderfmt{\glsgrouptitle{##1}}\indexspace}%  
}
```

`mcolindexhypergroup` The `mcolindexhypergroup` style is like the `mcolindexgroup` style but has hyper navigation.

```
\newglossarystyle{mcolindexhypergroup}{%
```

Base it on the `glostylemcolindex` style:

```
\setglossarystyle{mcolindex}%
```

Put navigation links to the groups at the start of the glossary:

```
\renewcommand*{\glossaryheader}{%
\item\glstreenavigationfmt{\glsnavigation}\indexspace}%
```

Add a heading for each group (with a target). The group's title is in bold followed by a vertical gap.

```
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
\indexspace}%
}
```

`mcolindexspannav` Similar to `mcolindexhypergroup`, but puts the navigation line in the optional argument of `multicols`.

```
\newglossarystyle{mcolindexspannav}{%
```

```
\setglossarystyle{index}%
```

```
\renewenvironment{theglossary}%
```

```
{%
```

```
\begin{multicols}{\glsncols}[\noindent\glstreenavigationfmt{\glsnavigation}]
```

```
\setlength{\parindent}{0pt}%
```

```
\setlength{\parskip}{0pt plus 0.3pt}%
```

```
\let\item\glstreeitem}%
```

```
{\end{multicols}}%
```

Add a heading for each group (with a target). The group's title is in bold followed by a vertical gap.

```
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
\indexspace}%
}
```

`mcoltree` Multi-column index style. Same as the tree, but puts the glossary in multiple columns.

```
\newglossarystyle{mcoltree}{%
```

```
\setglossarystyle{tree}%
```

```
\renewenvironment{theglossary}%
```

```
{%
```

```
\begin{multicols}{\glsncols}
```

```
\setlength{\parindent}{0pt}%
```

```
\setlength{\parskip}{0pt plus 0.3pt}%
```

```

    }%
    {\end{multicols}}%
  }

```

`mcoltreegroup` Like the `mcoltree` style but the glossary groups have headings.

```

\newglossarystyle{mcoltreegroup}{%
Base it on the glostylemcoltree style:
\setglossarystyle{mcoltree}%
Each group has a heading (in bold) followed by a vertical gap):
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}

```

`mcoltreehypergroup` The `mcoltreehypergroup` style is like the `treegroup` style, but has a set of links to the groups at the start of the glossary.

```

\newglossarystyle{mcoltreehypergroup}{%
Base it on the glostylemcoltree style:
\setglossarystyle{mcoltree}%
Put navigation links to the groups at the start of the theglossary environment:
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
Each group has a heading (in bold with a target) followed by a vertical gap):
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}

```

`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}{\glsncols}[\noindent\glstreenavigationfmt{\glsnavigation}]
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%

```

Each group has a heading (in bold with a target) followed by a vertical gap):

```

\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}

```

`mcoltreename` Multi-column index style. Same as the `treename`, but puts the glossary in multiple columns.

```
\newglossarystyle{mcoltreename}{%
  \setglossarystyle{treename}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}
```

`mcoltreenamegroup` Like the `mcoltreename` style but the glossary groups have headings.

```
\newglossarystyle{mcoltreenamegroup}{%
Base it on the glostylemcoltreename style:
  \setglossarystyle{mcoltreename}%
Give each group a heading:
  \renewcommand{\glsgroupeheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsggetgrouptitle{##1}}\par\indexspace}%
}
```

`mcoltreenamehypergroup` The `mcoltreenamehypergroup` style is like the `mcoltreenamegroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{mcoltreenamehypergroup}{%
Base it on the glostylemcoltreename style:
  \setglossarystyle{mcoltreename}%
Put navigation links to the groups at the start of the theglossary environment:
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
Each group has a heading (in bold with a target) followed by a vertical gap):
  \renewcommand*{\glsgroupeheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsggetgrouptitle{##1}}}\par
    \indexspace}%
}
```

`mcoltreenamepannav` Similar to the `mcoltreenamehypergroup` style but the navigation line is put in the optional argument of the `multicols` environment.

```
\newglossarystyle{mcoltreenamepannav}{%
  \setglossarystyle{treename}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
  }
```

```

        \setlength{\parskip}{0pt plus 0.3pt}%
    }%
    {\end{multicols}}%

```

Each group has a heading (in bold with a target) followed by a vertical gap):

```

    \renewcommand*{\glsgroupheading}[1]{%
        \par\noindent
        \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
        \indexspace}%
    }

```

mcolalttree Multi-column index style. Same as the `alttree`, but puts the glossary in multiple columns.

```

\newglossarystyle{mcolalttree}{%
    \setglossarystyle{alttree}%
    \renewenvironment{theglossary}%
    {%
        \begin{multicols}{\glsncols}
        \def\@gls@prevlevel{-1}%
        \mbox{}\par
    }%
    {\par\end{multicols}}%
}

```

mcolalttreegroup Like the `mcolalttree` style but the glossary groups have headings.

```

\newglossarystyle{mcolalttreegroup}{%
Base it on the glostylemcolalttree style:
    \setglossarystyle{mcolalttree}%
Give each group a heading.
    \renewcommand{\glsgroupheading}[1]{\par
        \def\@gls@prevlevel{-1}%
        \hangindent0pt\relax
        \parindent0pt\relax
        \glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
    }

```

mcolalttreehypergroup The `mcolalttreehypergroup` style is like the `mcolalttreegroup` style, but has a set of links to the groups at the start of the glossary.

```

\newglossarystyle{mcolalttreehypergroup}{%
Base it on the glostylemcolalttree style:
    \setglossarystyle{mcolalttree}%
Put the navigation links in the header
    \renewcommand*{\glossaryheader}{%
        \par
        \def\@gls@prevlevel{-1}%
        \hangindent0pt\relax
        \parindent0pt\relax
        \glstreenavigationfmt{\glsnavigation}\par\indexspace}%
    }

```

Put a hypertarget at the start of each group

```
\renewcommand*{\glsgroupheading}[1]{%
  \par
  \def\@gls@prevlevel{-1}%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}
```

`mcolalttreespannav` Similar to the `mcolalttreehypergroup` style but the navigation line is put in the optional argument of the `multicols` environment.

```
\newglossarystyle{mcolalttreespannav}{%
  \setglossarystyle{alttree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glscols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
```

Put a hypertarget at the start of each group

```
\renewcommand*{\glsgroupheading}[1]{%
  \par
  \def\@gls@prevlevel{-1}%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}
```

3.8 Glossary Styles using supertabular environment (`glossary-super.sty`)

The glossary styles defined in the `glossary-super` package use the `supertabular` environment.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-super-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-super-2021-11-01.sty}
\DeclareCurrentRelease{v4.53}{2023-09-29}
```

Declare package:

```
\ProvidesPackage{glossary-super}[2023/09/29 v4.53 (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*{\glsgruppeheading}[1]{}%
```

Main (level 0) entries put in a row (name in first column, description and page list in second column):

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
}%
```

Sub entries put in a row (no name, description and page list in second column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
  ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
```

```

        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{& \tabularnewline}%
    \fi
}

```

superborder The superborder style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{superborder}{%
```

Base it on the `glostylesuper` style:

```
\setglossarystyle{super}%
```

Put the glossary in a `supertabular` environment with two columns and a horizontal line in the head and tail:

```

\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  \begin{supertabular}{|l|p{\glsdescwidth}|}%
  {\end{supertabular}}%
}

```

superheader The superheader style is like the `super` style, but with a header:

```
\newglossarystyle{superheader}{%
```

Base it on the `glostylesuper` style:

```
\setglossarystyle{super}%
```

Put the glossary in a `supertabular` environment with two columns, a header and no tail:

```

\renewenvironment{theglossary}%
  {\tablehead{\bfseries \entryname &
  \bfseries \descriptionname \tabularnewline}%
  \tabletail{}%
  \begin{supertabular}{lp{\glsdescwidth}}%
  {\end{supertabular}}%
}

```

superheaderborder The superheaderborder style is like the `super` style but with a header and border:

```
\newglossarystyle{superheaderborder}{%
```

Base it on the `glostylesuper` style:

```
\setglossarystyle{super}%
```

Put the glossary in a `supertabular` environment with two columns, a header and horizontal lines above and below the table:

```

\renewenvironment{theglossary}%
  {\tablehead{\hline\bfseries \entryname &
  \bfseries \descriptionname \tabularnewline \hline}%
  \tabletail{\hline}
  \begin{supertabular}{|l|p{\glsdescwidth}|}%
  {\end{supertabular}}%
}

```


`super3col` The `super3col` style is like the `super` style, but with 3 columns:

```
\newglossarystyle{super3col}{%
```

Put the glossary in a `supertabular` environment with three columns and no head or tail:

```
\renewenvironment{theglossary}%  
  {\tablehead{}}\tabletail{}}%  
  \begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}%  
  {\end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgrupheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
\renewcommand{\glossentry}[2]{%  
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &  
  \glossentrydesc{##1} & ##2\tabularnewline  
}%
```

Sub entries on a row (no name, description in second column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%  
  &  
  \glssubentryitem{##2}%  
  \glstarget{##2}{\strut}\glossentrydesc{##2} &  
  ##3\tabularnewline  
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip  
  \renewcommand*{\glsgroupskip}{}%  
\else  
  \renewcommand*{\glsgroupskip}{& & \tabularnewline}%  
\fi  
}
```

`super3colborder` The `super3colborder` style is like the `super3col` style, but with a border:

```
\newglossarystyle{super3colborder}{%
```

Base it on the `glostylesuper3col` style:

```
\setglossarystyle{super3col}%
```

Put the glossary in a `supertabular` environment with three columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%  
  {\tablehead{\hline}\tabletail{\hline}}%
```

```

        \begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
        {\end{supertabular}}%
    }

```

super3colheader The `super3colheader` style is like the `super3col` style but with a header row:

```
\newglossarystyle{super3colheader}{%
```

Base it on the `glostylesuper3col` style:

```
\setglossarystyle{super3col}%
```

Put the glossary in a `supertabular` environment with three columns, a header and no tail:

```

\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline}\tabletail{}}%
  {\begin{supertabular}{|p{\glsdescwidth}p{\glspagelistwidth}|}%
  {\end{supertabular}}%
}

```

super3colheaderborder The `super3colheaderborder` style is like the `super3col` style but with a header and border:

```
\newglossarystyle{super3colheaderborder}{%
```

Base it on the `glostylesuper3colborder` style:

```
\setglossarystyle{super3colborder}%
```

Put the glossary in a `supertabular` environment with three columns, a header with horizontal lines and a horizontal line in the tail:

```

\renewenvironment{theglossary}%
  {\tablehead{\hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline}%
  {\tabletail{\hline}%
  {\begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
  {\end{supertabular}}%
}

```

super4col The `super4col` glossary style has four columns, where the third column contains the value of the corresponding symbol key used when that entry was defined.

```
\newglossarystyle{super4col}{%
```

Put the glossary in a `supertabular` environment with four columns and no head or tail:

```

\renewenvironment{theglossary}%
  {\tablehead{} \tabletail{}}%
  {\begin{supertabular}{|l|l|l|l|}%
  {\end{supertabular}}%

```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*\glsgroupheading}[1]{}
```

Main (level 0) entries on a row with the name in the first column, description in second column, symbol in third column and page list in last column:

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  \glossentrysymbol{##1} & ##2\tabularnewline
}%
```

Sub entries on a row with no name, the description in the second column, symbol in third column and page list in last column:

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  \glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip}{& & \tabularnewline}%
\fi
}
```

`super4colheader` The `super4colheader` style is like the `super4col` but with a header row.

```
\newglossarystyle{super4colheader}{%
```

Base it on the `glostylesuper4col` style:

```
\setglossarystyle{super4col}%
```

Put the glossary in a `supertabular` environment with four columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
    \bfseries\symbolname &
    \bfseries\pagelistname\tabularnewline}%
  \tabletail{}%
  \begin{supertabular}{l111}}%
  {\end{supertabular}}%
}
```

`super4colborder` The `super4colborder` style is like the `super4col` but with a border.

```
\newglossarystyle{super4colborder}{%
```

Base it on the `glostylesuper4col` style:

```
\setglossarystyle{super4col}%
```

Put the glossary in a `supertabular` environment with four columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  \begin{supertabular}{|1|1|1|1|}%
  {\end{supertabular}}%
}
```

`super4colheaderborder` The `super4colheaderborder` style is like the `super4col` but with a header and border.

```
\newglossarystyle{super4colheaderborder}{%
```

Base it on the `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{\glsdescwidth}lp{\glspagelistwidth}}}%
        {\end{supertabular}}%
    }

```

altsuper4colborder The `altsuper4colborder` style is like the `altsuper4col` but with a border.

```
\newglossarystyle{altsuper4colborder}{%
```

Base it on the `glostylesuper4colborder` style:

```
\setglossarystyle{super4colborder}%
```

Put the glossary in a `supertabular` environment with four columns and a horizontal line in the head and tail:

```

    \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}%
    {l|lp{\glsdescwidth}|l|p{\glspagelistwidth}|}}%
    {\end{supertabular}}%
}

```

altsuper4colheaderborder The `altsuper4colheaderborder` style is like the `altsuper4col` but with a header and border.

```
\newglossarystyle{altsuper4colheaderborder}{%
```

Base it on the `glostylesuper4colheaderborder` style:

```
\setglossarystyle{super4colheaderborder}%
```

Put the glossary in a `supertabular` environment with four columns and a header bordered by horizontal lines and a horizontal line in the tail:

```

\renewenvironment{theglossary}%
{\tablehead{\hline
  \bfseries\entryname &
  \bfseries\descriptionname &
  \bfseries\symbolname &
  \bfseries\pagelistname\tabularnewline\hline}%
\tabletail{\hline}%
\begin{supertabular}%
  {l|lp{\glsdescwidth}|l|p{\glspagelistwidth}|}}%
{\end{supertabular}}%
}

```

3.9 Glossary Styles using `supertabular` environment (`glossary-superragged.sty`)

The glossary styles defined in the `glossary-superragged` package use the `supertabular` environment. These styles are like those provided by the `glossary-super` package, except that the multiline columns have ragged right justification.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-superragged-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-superragged-2021-11-01.sty}
\DeclareCurrentRelease{v4.53}{2023-09-29}
```

Declare package:

```
\ProvidesPackage{glossary-superragged}[2023/09/29 v4.53 (NLCT)]
```

Requires the array package:

```
\RequirePackage{array}
```

Requires the supertabular package:

```
\RequirePackage{supertabular}
```

`\glsdescwidth` This is a length that governs the width of the description column. This may already have been defined.

```
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{}
```

`\glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined.

```
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{}
```

`superragged` The superragged glossary style uses the supertabular environment.

```
\newglossarystyle{superragged}{%
```

Put the glossary in a supertabular environment with two columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{}\tabletail{}%
  \begin{supertabular}{1>{\raggedright}p{\glsdescwidth}}%
  {\end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries put in a row (name in first column, description and page list in second column):

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1}\glspostdescription\space ##2%
  \tabularnewline
}%
```

Sub entries put in a row (no name, description and page list in second column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
  ##3%
  \tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip{& \tabularnewline}%
\fi
}
```

`superraggedborder` The `superraggedborder` style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{superraggedborder}{%
```

Base it on the `glostylesuperragged` style:

```
\setglossarystyle{superragged}%
```

Put the glossary in a `supertabular` environment with two columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|}%
  {\end{supertabular}}%
}
```

`superraggedheader` The `superraggedheader` style is like the `super` style, but with a header:

```
\newglossarystyle{superraggedheader}{%
```

Base it on the `glostylesuperragged` style:

```
\setglossarystyle{superragged}%
```

Put the glossary in a `supertabular` environment with two columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries \entryname & \bfseries \descriptionname
  \tabularnewline}%
  \tabletail{}%
  \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}}%
  {\end{supertabular}}%
}
```

`superraggedheaderborder` The `superraggedheaderborder` style is like the `superragged` style but with a header and border:

```
\newglossarystyle{superraggedheaderborder}{%
```

Base it on the `glostylesuper` style:

```
\setglossarystyle{superragged}%
```

Put the glossary in a `supertabular` environment with two columns, a header and horizontal lines above and below the table:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline\bfseries \entryname &
    \bfseries \descriptionname\tabularnewline\hline}%
  \tabletail{\hline}
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|}%
  \end{supertabular}}%
}
```

`superragged3col` The `superragged3col` style is like the `superragged` style, but with 3 columns:

```
\newglossarystyle{superragged3col}{%
```

Put the glossary in a `supertabular` environment with three columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{}\tabletail{}}%
  \begin{supertabular}{|l>{\raggedright}p{\glsdescwidth}%
    >{\raggedright}p{\glspagelistwidth}}%
  \end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  ##2\tabularnewline
}%
```

Sub entries on a row (no name, description in second column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  ##3\tabularnewline
}%
```


Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip{ }%
\else
  \renewcommand*\glsgroupskip{& & \tabularnewline}%
\fi
}
```

`superragged3colborder` The `superragged3colborder` style is like the `superragged3col` style, but with a border:

```
\newglossarystyle{superragged3colborder}{%
```

Base it on the `glostylesuperragged3col` style:

```
\setglossarystyle{superragged3col}%
```

Put the glossary in a `supertabular` environment with three columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
  >{\raggedright}p{\glspagelistwidth}|}%
  {\end{supertabular}}%
}
```

`superragged3colheader` The `superragged3colheader` style is like the `superragged3col` style but with a header row:

```
\newglossarystyle{superragged3colheader}{%
```

Base it on the `glostylesuperragged3col` style:

```
\setglossarystyle{superragged3col}%
```

Put the glossary in a `supertabular` environment with three columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
  \bfseries\pagelistname\tabularnewline}\tabletail{ }%
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}%
  >{\raggedright}p{\glspagelistwidth}}%
  {\end{supertabular}}%
}
```

`raggedright3colheaderborder` The `superragged3colheaderborder` style is like the `superragged3col` style but with a header and border:

```
\newglossarystyle{superragged3colheaderborder}{%
```

Base it on the `glostylesuperragged3colborder` style:

```
\setglossarystyle{superragged3colborder}%
```

Put the glossary in a supertabular environment with three columns, a header with horizontal lines and a horizontal line in the tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline}%
  \tabletail{\hline}%
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
    >{\raggedright}p{\glspagelistwidth}|}%
  \end{supertabular}}%
}
```

`altsuperragged4col` The `altsuperragged4col` glossary style is like `altsuper4col` style in the `glossary-super` package but uses ragged right formatting in the description and page list columns.

```
\newglossarystyle{altsuperragged4col}{%
```

Put the glossary in a supertabular environment with four columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{}\tabletail{}%
  \begin{supertabular}{|l>{\raggedright}p{\glsdescwidth}l%
    >{\raggedright}p{\glspagelistwidth}|}%
  \end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row with the name in the first column, description in second column, symbol in third column and page list in last column:

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  \glossentrysymbol{##1} & ##2\tabularnewline
}%
```

Sub entries on a row with no name, the description in the second column, symbol in third column and page list in last column:

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  \glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```

\ifglsgroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi
}

```

`altsuperragged4colheader` The `altsuperragged4colheader` style is like the `altsuperragged4col` style but with a header row.

```
\newglossarystyle{altsuperragged4colheader}{%
```

Base it on the `glostylealtsuperragged4col` style:

```
\setglossarystyle{altsuperragged4col}%
```

Put the glossary in a `supertabular` environment with four columns, a header and no tail:

```

\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
    \bfseries\symbolname &
    \bfseries\pagelistname\tablearnewline}\tabletail{}}%
  {\begin{supertabular}{1>{\raggedright}p{\glsgdescwidth}1%
    >{\raggedright}p{\glspagelistwidth}}}%
  {\end{supertabular}}%
}

```

`altsuperragged4colborder` The `altsuperragged4colborder` style is like the `altsuperragged4col` style but with a border.

```
\newglossarystyle{altsuperragged4colborder}{%
```

Base it on the `glostylealtsuperragged4col` style:

```
\setglossarystyle{altsuper4col}%
```

Put the glossary in a `supertabular` environment with four columns and a horizontal line in the head and tail:

```

\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  \begin{supertabular}%
    {1|>{\raggedright}p{\glsgdescwidth}1|1%
    >{\raggedright}p{\glspagelistwidth}1}}%
  {\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}%
    {1|>{\raggedright}p{\glsdescwidth}|1|}%
    >{\raggedright}p{\glspagelistwidth}|}%
  \end{supertabular}}%
}

```

3.10 Tree Styles (`glossary-tree.sty`)

The `glossary-tree` style file defines glossary styles that have a tree-like structure. These are designed for hierarchical glossaries.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```

\DeclareRelease{v4.46}{2020-03-19}{glossary-tree-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-tree-2021-11-01.sty}
\DeclareCurrentRelease{v4.53}{2023-09-29}

```

Declare package:

```
\ProvidesPackage{glossary-tree}[2023/09/29 v4.53 (NLCT)]
```

`\indexspace` There are a few classes that don't define `\indexspace`, so provide a definition if it hasn't been defined.

```

\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}

```

`\glstreenamefmt` Format used to display the name in the tree styles. (This may be counteracted by `\glsnamefont`.) This command was previously also used to format the group headings.

```
\newcommand*{\glstreenamefmt}[1]{\textbf{#1}}
```

`\glstreegroupheaderfmt` Format used to display the group header in the tree styles. Before v4.22, `\glstreenamefmt` was used for the group header, so the default definition uses that to help maintain backward-compatibility, since in previous versions redefining `\glstreenamefmt` would've also affected the group headings.

```
\newcommand*{\glstreegroupheaderfmt}[1]{\glstreenamefmt{#1}}
```

`\glstreenavigationfmt` Format used to display the navigation header in the tree styles.

```
\newcommand*{\glstreenavigationfmt}[1]{\glstreenamefmt{#1}}
```

Allow the user to adjust the index style without disturbing the index.

`\glstreeitem` Top level item used in index style.

```
\ifdef\@idxitem
{\newcommand{\glstreeitem}{\@idxitem}}
{\newcommand{\glstreeitem}{\par\hangindent40\p@}}
```

`\glstreesubitem` Level 1 item used in index style.

```
\ifdef\subitem
{\let\glstreesubitem\subitem}
{\newcommand\glstreesubitem{\glstreeitem\hspace*{20\p@}}}
```

`\glstreesubsubitem` Level 1 item used in index style.

```
\ifdef\subsubitem
{\let\glstreesubsubitem\subsubitem}
{\newcommand\glstreesubsubitem{\glstreeitem\hspace*{30\p@}}}
```

`\glstreepredesc` Allow the user to adjust the space before the description (except for the `alttree` style).

```
\newcommand{\glstreepredesc}{\space}
```

`\glstreechildpredesc` Allow the user to adjust the space before the description for sub-entries (except for the `treenoname` and `alttree` style).

```
\newcommand{\glstreechildpredesc}{\space}
```

`index` The index glossary style is similar in style to the way indices are usually typeset using `\item`, `\subitem` and `\subsubitem`. The entry name is set in bold. If an entry has a symbol, it is placed in brackets after the name. Then the description is displayed, followed by the number list. This style allows up to three levels.

```
\newglossarystyle{index}{%
```

Set the paragraph indentation and skip and define `\item` to be the same as that used by `theindex`:

```
\renewenvironment{theglossary}%
{\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
\let\item\glstreeitem
\let\subitem\glstreesubitem
\let\subsubitem\glstreesubsubitem
}%
{\par}%
```

Do nothing at the start of the environment:

```
\renewcommand*{\glossaryheader}{}%
```

No group headers:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entry starts a new item with the name in bold followed by the symbol in brackets (if it exists), the description and the page list.

```
\renewcommand*{\glossentry}[2]{%
  \item\glstreeitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
  \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
  \glstreepredesc \glossentrydesc{##1}\glspostdescription\space ##2%
}%
```

Sub entries: level 1 entries use `\subitem`, levels greater than 1 use `\subsubitem`. The level (`##1`) shouldn't be 0, as that's catered by `\glossentry`, but for completeness, if the level is 0, `\item` is used. The name is put in bold, followed by the symbol in brackets (if it exists), the description and the page list.

```
\renewcommand{\subglossentry}[3]{%
  \ifcase##1\relax
    % level 0
    \item
  \or
    % level 1
    \subitem
    \glssubentryitem{##2}%
  \else
    % all other levels
    \subsubitem
  \fi
  \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
  \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
  \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3%
}%
```

Vertical gap between groups is the same as that used by indices:

```
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
```

`indexgroup` The `indexgroup` style is like the `index` style but has headings.

```
\newglossarystyle{indexgroup}{%
```

Base it on the `glostyleindex` style:

```
\setglossarystyle{index}%
```

Add a heading for each group. This puts the group's title in bold followed by a vertical gap.

```
\renewcommand*{\glsgroupheading}[1]{%
  \item\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
  \indexspace
}%
}
```

`indexhypergroup` The `indexhypergroup` style is like the `indexgroup` style but has hyper navigation.

```
\newglossarystyle{indexhypergroup}{%
```

Base it on the `glostyleindex` style:

```
\setglossarystyle{index}%
```

Put navigation links to the groups at the start of the glossary:

```
\renewcommand*\glossaryheader}{%
  \item\glstreenavigationfmt{\glsnavigation}\indexspace}%
```

Add a heading for each group (with a target). The group's title is in bold followed by a vertical gap.

```
\renewcommand*\glsgroupheading}[1]{%
  \item\glstreegroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
  \indexspace}%
}
```

tree The tree glossary style is similar in style to the index style, but can have arbitrary levels.

```
\newglossarystyle{tree}{%
```

Set the paragraph indentation and skip:

```
\renewenvironment{theglossary}%
  {\setlength{\parindent}{0pt}%
   \setlength{\parskip}{0pt plus 0.3pt}}%
  {}%
```

Do nothing at the start of the theglossary environment:

```
\renewcommand*\glossaryheader}{}%
```

No group headings:

```
\renewcommand*\glsgroupheading}[1]{}%
```

Main (level 0) entries: name in bold, followed by symbol in brackets (if it exists), the description and the page list:

```
\renewcommand{\glossentry}[2]{%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
  \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
  \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
}%
```

Sub entries: level $\langle n \rangle$ is indented by $\langle n \rangle$ times `\glstreeindent`. The name is in bold, followed by the symbol in brackets (if it exists), the description and the page list.

```
\renewcommand{\subglossentry}[3]{%
  \hangindent##1\glstreeindent\relax
  \parindent##1\glstreeindent\relax
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
  \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
  \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3\par
}%
```

Vertical gap between groups is the same as that used by indices:

```
\renewcommand*{\glsgroupskip}{\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{\glsgetgrouptitle{##1}}\par
\indexspace}%
}
```

treehypergroup The treehypergroup style is like the treegroup style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{treehypergroup}{%
```

Base it on the glostyletree style:

```
\setglossarystyle{tree}%
```

Put navigation links to the groups at the start of the theglossary environment:

```
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
```

\glstreeindent Length governing left indent for each level of the tree style.

```
\newlength\glstreeindent
\setlength{\glstreeindent}{10pt}
```

treenoname The treenoname glossary style is like the tree style, but doesn't print the name or symbol for sub-levels.

```
\newglossarystyle{treenoname}{%
```

Set the paragraph indentation and skip:

```
\renewenvironment{theglossary}%
{\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}}%
{}
```

No header:

```
\renewcommand*{\glossaryheader}{}%
```


No group headings:

```
\renewcommand*\glsgroupheading}[1]{}
```

Main (level 0) entries: the name is in bold, followed by the symbol in brackets (if it exists), the description and the page list.

```
\renewcommand{\glossentry}[2]{%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
  \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
  \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
}%
```

Sub entries: level $\langle n \rangle$ is indented by $\langle n \rangle$ times `\glstreeindent`. The name and symbol are omitted. The description followed by the page list are displayed.

```
\renewcommand{\subglossentry}[3]{%
  \hangindent##1\glstreeindent\relax
  \parindent##1\glstreeindent\relax
  \ifnum##1=1\relax
  \glssubentryitem{##2}%
  \fi
  \glstarget{##2}{\strut}%
  \glossentrydesc{##2}\glspostdescription\space##3\par
}%
```

Vertical gap between groups is the same as that used by indices:

```
\renewcommand*\glsgroupskip{\ifglsnogroupskip\else\indexspace\fi}%
}
```

`treenonamegroup` Like the `treenoname` style but the glossary groups have headings.

```
\newglossarystyle{treenonamegroup}{%
```

Base it on the `glostyletreenoname` style:

```
\setglossarystyle{treenoname}%
```

Give each group a heading:

```
\renewcommand{\glsgroupheading}[1]{\par
  \noindent\glstreegroupheaderfmt
  {\glsgetgrouptitle{##1}}\par\indexspace}%
}
```

`treenonamehypergroup` The `treenonamehypergroup` style is like the `treenonamegroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{treenonamehypergroup}{%
```

Base it on the `glostyletreenoname` style:

```
\setglossarystyle{treenoname}%
```

Put navigation links to the groups at the start of the `theglossary` environment:

```
\renewcommand*\glossaryheader{%
  \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupeheading}[1]{%
  \par\noindent
  \glstreenameheaderfmt
  {\glslnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
  \indexspace}%
}
```

`\glswidestname` Find the widest name over all parentless entries in the given glossary or glossaries.

```
\newrobustcmd*{\glswidestname}[1][\glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  \forallglossaries[#1]{\gls@type}%
  {%
    \forallglsentries[\gls@type]{\glo@label}%
    {%
      \ifglshasparent{\glo@label}%
      {}%
      {%
        \glsmeasurewidth{\dimen@}%
        {\glstreenamefmt{\glsentryname{\glo@label}}}%
        \ifdim\dimen@>\gls@tmplen
          \gls@tmplen=\dimen@
          \letcs{\glswidestname}{glo@\glsdetoklabel{\glo@label}@name}%
        \fi
      }%
    }%
  }%
}
```

`\glsssetwidest` `\glsssetwidest[level]{text}` sets the widest text for the given level. It is used by the `alttree` glossary styles to determine the indentation of each level.

```
\newcommand*{\glsssetwidest}[2][0]{%
  \expandafter\def\csname @glswidestname\romannumeral#1\endcsname{%
  #2}%
}
```

`\@glswidestname` Initialise `\@glswidestname`.

```
\newcommand*{\@glswidestname}{}
```

`\glstreenamebox` Used by the `alttree` style to create the box for the name and associated information.

```
\newcommand*{\glstreenamebox}[2]{%
  \makebox[#1][l]{#2}%
}
```

`alttree` The `alttree` glossary style is similar in style to the `tree` style, but the indentation is obtained from the width of `\@glswidestname` which is set using `\glsssetwidest`.

```
\newglossarystyle{alttree}{%
```

Redefine theglossary environment.

```
\renewenvironment{theglossary}%  
  {\def\@gls@prevlevel{-1}%  
   \mbox{}\par}%  
  {\par}%
```

Set the header and group headers to nothing.

```
\renewcommand*\glossaryheader{}%  
\renewcommand*\glsgroupheading}[1]{}%
```

Redefine the way that the level 0 entries are displayed.

```
\renewcommand{\glossentry}[2]{%  
  \ifnum\@gls@prevlevel=0\relax  
  \else
```

Find out how big the indentation should be by measuring the widest entry.

```
  \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%  
  \fi
```

Set the hangindent and paragraph indent.

```
  \hangindent\glstreeindent  
  \parindent\glstreeindent
```

Put the name to the left of the paragraph block.

```
  \makebox[0pt][r]{\glstreenamebox{\glstreeindent}{%  
    \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}}%
```

If the symbol is missing, ignore it, otherwise put it in brackets.

```
  \ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}%
```

Do the description followed by the description terminator and location list.

```
  \glossentrydesc{##1}\glspostdescription \space ##2\par
```

Set the previous level to 0.

```
  \def\@gls@prevlevel{0}%  
  }%
```

Redefine the way sub-entries are displayed.

```
\renewcommand{\subglossentry}[3]{%
```

Increment and display the sub-entry counter if this is a level 1 entry and the sub-entry counter is in use.

```
  \ifnum##1=1\relax  
    \glssubentryitem{##2}%  
  \fi
```

If the level hasn't changed, keep the same settings, otherwise adjust `\glstreeindent` accordingly.

```
  \ifnum\@gls@prevlevel=##1\relax  
  \else
```

Compute the widest entry for this level, or for level 0 if not defined for this level.

Store in `\gls@tmplen`

```
\@ifundefined{@glswidestname\romannumeral##1}{%
  \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}{%
  \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{%
    \csname @glswidestname\romannumeral##1\endcsname\space}}}%
```

Determine if going up or down a level

```
\ifnum\@gls@prevlevel<##1\relax
```

Depth has increased, so add the width of the widest entry to `\glstreeindent`.

```
\setlength\glstreeindent\gls@tmplen
\addtolength\glstreeindent\parindent
\parindent\glstreeindent
\else
```

Depth has decreased, so subtract width of the widest entry from the previous level to `\glstreeindent`. First determine the width of the widest entry for the previous level and store in `\glstreeindent`.

```
\@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
  \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{%
    \@glswidestname\space}}{%
  \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{%
    \csname @glswidestname\romannumeral\@gls@prevlevel
    \endcsname\space}}}%
```

Subtract this length from the previous level's paragraph indent and set to `\glstreeindent`.

```
\addtolength\parindent{-\glstreeindent}%
\setlength\glstreeindent\parindent
\fi
\fi
```

Set the hanging indentation.

```
\hangindent\glstreeindent
```

Put the name to the left of the paragraph block

```
\makebox[Opt][r]{\glstreenamebox{\gls@tmplen}{%
  \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
```

If the symbol is missing, ignore it, otherwise put it in brackets.

```
\ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
```

Do the description followed by the description terminator and location list.

```
\glossentrydesc{##2}\glspostdescription\space ##3\par
```

Set the previous level macro to the current level.

```
\def\@gls@prevlevel{##1}%
}%
```

Vertical gap between groups is the same as that used by indices:

```
\renewcommand*{\glsgroupskip}{\ifglsgroupskip\else\indexspace\fi}%
}
```

`almtreegroup` Like the `almtree` style but the glossary groups have headings.

```
\newglossarystyle{almtreegroup}{%
Base it on the glostylealmtree style:
\setglossarystyle{almtree}%
Give each group a heading.
\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
\par\indexspace}%
}
```

`almtreehypergroup` The `almtreehypergroup` style is like the `almtreegroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{almtreehypergroup}{%
Base it on the glostylealmtree style:
\setglossarystyle{almtree}%
Put the navigation links in the header
\renewcommand*{\glossaryheader}{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
Put a hypertext at the start of each group
\renewcommand*{\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}}
```

4 Multi-Lingual Support

4.1 Babel Captions (`glossaries-babel.sty`)

Many thanks to everyone who contributed to the translations both via email and on `comp.text.tex`. Language support has now been split off into independent language modules.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossaries-babel-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-babel-2021-11-01.sty}
\DeclareCurrentRelease{v4.53}{2023-09-29}
```

Declare package

```
\ProvidesPackage{glossaries-babel}[2023/09/29 v4.53 (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{v4.53}{2023-09-29}
```

Declare package

```
\ProvidesPackage{glossaries-polyglossia}[2023/09/29 v4.53 (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{v4.53}{2023-09-29}
```

Declare package:

```
\ProvidesPackage{glossaries-prefix}[2023/09/29 v4.53 (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@glstdetoklabel{#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\@glstr@record
  {\@glstr@record{#1}{#2}{glslink}}%
  {}%
}
```

`\glsprefixsep` Separator between prefix and term. Does nothing by default.

```
\newcommand{\glsprefixsep}{}
```

Define commands that insert the prefix before commands like `\gls`:

`\pgls`

```
\newrobustcmd{\pgls}{\@gls@hyp@opt\@pgls}
```

`\@pgls` Unstarred version.

```
\newcommand*{\@pgls}[2][{}]{%
  \new@ifnextchar[%
  {\@pgls@{#1}{#2}}%
  {\@pgls@{#1}{#2}[]}%
}
```

`\@pgls@` Read in the final optional argument:

```
\def\@pgls@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglshasprefix{#2}{\glsentryprefix{#2}\glsprefixsep}{}%
    }%
    {\ifglshasprefixfirst{#2}{\glsentryprefixfirst{#2}\glsprefixsep}{}%
    }%
  }%
  \@gls@{#1}{#2}[#3]%
}%
}
```

Similarly for the plural version:

```
\pglsp1
\newrobustcmd{\pglsp1}{\@gls@hyp@opt\@pglsp1}
```

```
\@pglsp1 Unstarred version.
\newcommand*{\@pglsp1}[2][{}]{%
\new@ifnextchar[%
{\@pglsp1@{#1}{#2}}%
{\@pglsp1@{#1}{#2}[]}%
}
```

\@pglsp1@ Read in the final optional argument:

```
\def\@pglsp1@#1#2[#3]{%
\@glsprefix@record@hook{#1}{#2}%
\glsdoifexists{#2}%
{%
\ifglsused{#2}%
{%
\ifgls@hasprefixplural{#2}{\glsentryprefixplural{#2}\glsprefixsep}{}%
}%
{%
\ifgls@hasprefixfirstplural{#2}%
{\glsentryprefixfirstplural{#2}\glsprefixsep}{}%
}%
\@glspl@{#1}{#2}[#3]%
}%
}
```

Now for the first letter upper case versions:

```
\PglS
\newrobustcmd{\PglS}{\@gls@hyp@opt\@PglS}
```

Add sentence case mapping.

```
\glsmfuaddmap{\pgls}{\PglS}
```

```
\@PglS Unstarred version.
\newcommand*{\@PglS}[2][{}]{%
\new@ifnextchar[%
{\@PglS@{#1}{#2}}%
{\@PglS@{#1}{#2}[]}%
}
```

\@PglS@ Read in the final optional argument:

```
\def\@PglS@#1#2[#3]{%
\@glsprefix@record@hook{#1}{#2}%
\glsdoifexists{#2}%
{%
\ifglsused{#2}%
}
```

```

{%
  \ifglshasprefix{#2}%
  {%
    \Glentryprefix{#2}%
    \glsprefixsep
    \@gls@{#1}{#2}[#3]%
  }%
  {\@Gls@{#1}{#2}[#3]}%
}%
{%
  \ifglshasprefixfirst{#2}%
  {%
    \Glentryprefixfirst{#2}%
    \glsprefixsep
    \@gls@{#1}{#2}[#3]%
  }%
  {\@Gls@{#1}{#2}[#3]}%
}%
}%
}

```

Similarly for the plural version:

```

\Pglsp1
\newrobustcmd{\Pglsp1}{\@gls@hyp@opt\Pglsp1}

```

Add sentence case mapping.

```

\glsmfuaddmap{\pglsp1}{\Pglsp1}

```

\@Pglsp1 Unstarred version.

```

\newcommand*{\@Pglsp1}[2][{}]{%
  \new@ifnextchar[%
    {\@Pglsp1@{#1}{#2}}%
    {\@Pglsp1@{#1}{#2}[{}]}%
}

```

\@Pglsp1@ Read in the final optional argument:

```

\def\@Pglsp1@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglshasprefixplural{#2}%
      {%
        \Glentryprefixplural{#2}%
        \glsprefixsep
        \@glspl@{#1}{#2}[#3]%
      }%
    }%
  }%
}

```

```

        {\@Glspl@{#1}{#2}[#3]}%
    }%
    {%
        \ifglshasprefixfirstplural{#2}%
        {%
            \Glsentryprefixfirstplural{#2}%
            \glsprefixsep
            \@glspl@{#1}{#2}[#3]%
        }%
        {\@Glspl@{#1}{#2}[#3]}%
    }%
}
}

```

Finally the all upper case versions:

```

\PGLS
\newrobustcmd{\PGLS}{\@gls@hyp@opt\@PGLS}

```

\@PGLS Unstarred version.

```

\newcommand*{\@PGLS}[2] [] {%
    \new@ifnextchar [%
        {\@PGLS@{#1}{#2}}%
        {\@PGLS@{#1}{#2} []}%
    }
}

```

\@PGLS@ Read in the final optional argument:

```

\def\@PGLS@#1#2[#3]{%
    \@glsprefix@record@hook{#1}{#2}%
    \glsdoifexists{#2}%
    {%
        \ifglsused{#2}%
        {%
            \ifglshasprefix{#2}%
            {\glsuppercase{\glsentryprefix{#2}\glsprefixsep}}{%
            }%
        }%
        {%
            \ifglshasprefixfirst{#2}%
            {\glsuppercase{\glsentryprefixfirst{#2}\glsprefixsep}}{%
            }%
        }%
        \@GLS@{#1}{#2}[#3]%
    }%
}
}

```

Plural version:

```

\PGLSpl
\newrobustcmd{\PGLSpl}{\@gls@hyp@opt\@PGLSpl}

```

```

\@PGLSp1 Unstarred version.
  \newcommand*{\@PGLSp1}[2] [] {%
    \new@ifnextchar [%
      {\@PGLSp1@{#1}{#2}}%
      {\@PGLSp1@{#1}{#2} []}%
    }

```

\@PGLSp1@ Read in the final optional argument:

```

\def\@PGLSp1@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglschasprefixplural{#2}%
        {\glsuppercase{\glsentryprefixplural{#2}\glsprefixsep}}{%}
      }%
    }%
    \ifglschasprefixfirstplural{#2}%
    {\glsuppercase{\glsentryprefixfirstplural{#2}\glsprefixsep}}{%}
  }%
  \@GLSp1@{#1}{#2}[#3]%
}%
}

```

6 Accessibility Support (glossaries-accsupp Code)

The glossaries-accsupp package is experimental. It is intended to provide a means of using the PDF accessibility support in glossary entries. See the accsupp documentation for further details about accessibility support.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```

\DeclareRelease{v4.46}{2020-03-19}{glossaries-accsupp-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-accsupp-2021-11-01.sty}
\DeclareCurrentRelease{v4.53}{2023-09-29}

```

Package version number now in line with main glossaries package number.

```

\ProvidesPackage{glossaries-accsupp}[2023/09/29 v4.53 (NLCT)
  Experimental glossaries accessibility]

```

Pass all options to glossaries:

```
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
```

Process options:

```
\ProcessOptions
```

This package should be loaded before glossaries-extra, so complain if that has already been loaded.

```

\@ifpackageloaded{glossaries-extra}
{%

```

If the `accsupp` option was used, `\@glsxtr@doaccsupp` will have been set, otherwise it will be empty.

```
\ifx\@glsxtr@doaccsupp\empty
  \GlossariesWarning{The ‘glossaries-accsupp’
package has been loaded\MessageBreak
after the ‘glossaries-extra’ package. This\MessageBreak
can cause a failure to integrate both packages. \MessageBreak
Either use the ‘accsupp’ option when you load\MessageBreak
‘glossaries-extra’ or load ‘glossaries-accsupp’\MessageBreak
before loading ‘glossaries-extra’}%
\fi
}
{}
```

`\compatibleglossentry` Override style compatibility macros:

`\compatiblesubglossentry`

Required packages:

```
\RequirePackage{glossaries}
```

`\gls@accsupp@engine` There’s currently only support for `accsupp`, but if you define `\gls@accsupp@engine` before loading `glossaries-accsupp`, you can prevent `accsupp` from being loaded. Redefining this command after `glossaries-accsupp` has loaded obviously won’t do anything (and so is an internal command to deter casual use). If it is defined to something other than `accsupp` then `\gls@accessibility` will need to be defined to something appropriate.

```
\providecommand{\gls@accsupp@engine}{accsupp}
```

```
\gls@accessibility[<options>]{<PDF element>}{<value>}
{<content>}
```

`\gls@accessibility`

```
\providecommand{\gls@accessibility}[4]{#4}
\ifdefstring\gls@accsupp@engine{accsupp}
{
  \RequirePackage{accsupp}
  \renewcommand{\gls@accessibility}[4]{%
    \BeginAccSupp{#1,#2={#3}}#4\EndAccSupp{}}%
  }
}
{}
```

```
\glsaccessibility[<options>]{<PDF element>}{<value>}
{<content>}
```

`\glsaccessibility`

User-level command that includes debug info if required.

```
\newcommand{\glsaccessibility}[4] []{%  
  \glsshowaccsupp{#1}{#2}{#3}%  
  \gls@accessibility{#1}{#2}{#3}{#4}%  
}
```

6.1 Defining Replacement Text

The version 0.1 stored the replacement text in the symbol key. This has been changed to use the new keys defined here. Example of use:

```
%\newglossaryentry{dr}{name=Dr,description={},access={Doctor}}  
%
```

access The replacement text corresponding to the name key:

```
\define@key{glossentry}{access}{%  
  \def\@glo@access{#1}%  
}
```

textaccess The replacement text corresponding to the text key:

```
\define@key{glossentry}{textaccess}{%  
  \def\@glo@textaccess{#1}%  
}
```

firstaccess The replacement text corresponding to the first key:

```
\define@key{glossentry}{firstaccess}{%  
  \def\@glo@firstaccess{#1}%  
}
```

pluralaccess The replacement text corresponding to the plural key:

```
\define@key{glossentry}{pluralaccess}{%  
  \def\@glo@pluralaccess{#1}%  
}
```

firstpluralaccess The replacement text corresponding to the firstplural key:

```
\define@key{glossentry}{firstpluralaccess}{%  
  \def\@glo@firstpluralaccess{#1}%  
}
```

symbolaccess The replacement text corresponding to the symbol key:

```
\define@key{glossentry}{symbolaccess}{%  
  \def\@glo@symbolaccess{#1}%  
}
```

symbolpluralaccess The replacement text corresponding to the symbolplural key:

```
\define@key{glossentry}{symbolpluralaccess}{%  
  \def\@glo@symbolpluralaccess{#1}%  
}
```


descriptionaccess The replacement text corresponding to the description key:

```
\define@key{glossentry}{descriptionaccess}{%  
  \def\@glo@descaccess{#1}%  
}
```

descriptionpluralaccess The replacement text corresponding to the descriptionplural key:

```
\define@key{glossentry}{descriptionpluralaccess}{%  
  \def\@glo@descpluralaccess{#1}%  
}
```

shortaccess The replacement text corresponding to the short key:

```
\define@key{glossentry}{shortaccess}{%  
  \def\@glo@shortaccess{#1}%  
}
```

shortpluralaccess The replacement text corresponding to the shortplural key:

```
\define@key{glossentry}{shortpluralaccess}{%  
  \def\@glo@shortpluralaccess{#1}%  
}
```

longaccess The replacement text corresponding to the long key:

```
\define@key{glossentry}{longaccess}{%  
  \def\@glo@longaccess{#1}%  
}
```

longpluralaccess The replacement text corresponding to the longplural key:

```
\define@key{glossentry}{longpluralaccess}{%  
  \def\@glo@longpluralaccess{#1}%  
}
```

There are now also keys that correspond to the user keys:

user1access The replacement text corresponding to the user1 key:

```
\define@key{glossentry}{user1access}{%  
  \def\@glo@useriaccess{#1}%  
}
```

user2access The replacement text corresponding to the user2 key:

```
\define@key{glossentry}{user2access}{%  
  \def\@glo@useriiaccess{#1}%  
}
```

user3access The replacement text corresponding to the user3 key:

```
\define@key{glossentry}{user3access}{%  
  \def\@glo@useriiiaccess{#1}%  
}
```

`user4access` The replacement text corresponding to the `user4` key:

```
\define@key{glossentry}{user4access}{%
  \def\@glo@userivaccess{#1}%
}
```

`user5access` The replacement text corresponding to the `user5` key:

```
\define@key{glossentry}{user5access}{%
  \def\@glo@uservaccess{#1}%
}
```

`user6access` The replacement text corresponding to the `user6` key:

```
\define@key{glossentry}{user6access}{%
  \def\@glo@userviaccess{#1}%
}
```

For any custom keys, the replacement text would have to be explicitly put in the value, e.g., `user1={\glsshortaccsupp{inches}{in}}`.

Append these new keys to `\@gls@keymap`:

```
\appto\@gls@keymap{,%
  {access}{access},%
  {textaccess}{textaccess},%
  {firstaccess}{firstaccess},%
  {pluralaccess}{pluralaccess},%
  {firstpluralaccess}{firstpluralaccess},%
  {symbolaccess}{symbolaccess},%
  {symbolpluralaccess}{symbolpluralaccess},%

  {descriptionaccess}{descaccess},%
  {descriptionpluralaccess}{descpluralaccess},%
  {shortaccess}{shortaccess},%
  {shortpluralaccess}{shortpluralaccess},%
  {longaccess}{longaccess},%
  {longpluralaccess}{longpluralaccess},%
  {user1access}{useriaccess},%
  {user2access}{useriiaccess},%
  {user3access}{useriiiaccess},%
  {user4access}{userivaccess},%
  {user5access}{uservaccess},%
  {user6access}{userviaccess}%
}
```

`\@gls@noaccess` Indicates that no replacement text has been provided.

```
\def\@gls@noaccess{\relax}
```

Previously, the access key was initialised to the value of the `symbol` key at the start for backwards compatibility. This causes a problem for situations where the replacement text is provided for `symbol` but not for `name` so this behaviour has been removed.

```
\let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
```

```

\renewcommand*{\@newglossaryentryprehook}{%
  \@gls@oldnewglossaryentryprehook
  \def\@glo@access{\relax}%

```

Initialise the other keys:

```

\def\@glo@textaccess{\@glo@access}%
\def\@glo@firstaccess{\@glo@access}%
\def\@glo@pluralaccess{\@glo@textaccess}%
\def\@glo@firstpluralaccess{\@glo@pluralaccess}%
\def\@glo@symbolaccess{\relax}%
\def\@glo@symbolpluralaccess{\@glo@symbolaccess}%
\def\@glo@descaccess{\relax}%
\def\@glo@descpluralaccess{\@glo@descaccess}%
\def\@glo@shortaccess{\relax}%
\def\@glo@shortpluralaccess{\@glo@shortaccess}%
\def\@glo@longaccess{\relax}%
\def\@glo@longpluralaccess{\@glo@longaccess}%
\def\@glo@useriaccess{\relax}%
\def\@glo@useriiaaccess{\relax}%
\def\@glo@useriiaaaccess{\relax}%
\def\@glo@userivaccess{\relax}%
\def\@glo@uservaccess{\relax}%
\def\@glo@userviaccess{\relax}%
}

```

Add to the end hook:

```

\let\@gls@oldnewglossaryentryposthook\@newglossaryentryposthook
\renewcommand*{\@newglossaryentryposthook}{%
  \@gls@oldnewglossaryentryposthook

```

Store the access information:

```

\expandafter
\protected@xdef\csname glo@\@glo@label @access\endcsname{%
  \@glo@access}%
\expandafter
\protected@xdef\csname glo@\@glo@label @textaccess\endcsname{%
  \@glo@textaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstaccess\endcsname{%
  \@glo@firstaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @pluralaccess\endcsname{%
  \@glo@pluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstpluralaccess\endcsname{%
  \@glo@firstpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolaccess\endcsname{%
  \@glo@symbolaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolpluralaccess\endcsname{%

```

```

    \@glo@symbolpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descaccess\endcsname{%
  \@glo@descaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descpluralaccess\endcsname{%
  \@glo@descpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortaccess\endcsname{%
  \@glo@shortaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortpluralaccess\endcsname{%
  \@glo@shortpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longaccess\endcsname{%
  \@glo@longaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longpluralaccess\endcsname{%
  \@glo@longpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriaccess\endcsname{%
  \@glo@useriaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiaccess\endcsname{%
  \@glo@useriiaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiiaccess\endcsname{%
  \@glo@useriiiaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @userivaccess\endcsname{%
  \@glo@userivaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @uservaccess\endcsname{%
  \@glo@uservaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @userviaccess\endcsname{%
  \@glo@userviaccess}%
}

```

6.2 Accessing Replacement Text

`\glsentryaccess` Get the value of the access key for the entry with the given label:

```

\newcommand*{\glsentryaccess}[1]{%
  \@gls@entry@field{#1}{access}%
}

```

`\glsentrytextaccess` Get the value of the textaccess key for the entry with the given label:

```

\newcommand*{\glsentrytextaccess}[1]{%
  \@gls@entry@field{#1}{textaccess}%
}

```

}

`\glsentryfirstaccess` Get the value of the `firstaccess` key for the entry with the given label:

```
\newcommand*{\glsentryfirstaccess}[1]{%
  \@gls@entry@field{#1}{firstaccess}%
}
```

`\glsentrypluralaccess` Get the value of the `pluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentrypluralaccess}[1]{%
  \@gls@entry@field{#1}{pluralaccess}%
}
```

`\glsentryfirstpluralaccess` Get the value of the `firstpluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentryfirstpluralaccess}[1]{%
  \@gls@entry@field{#1}{firstpluralaccess}%
}
```

`\glsentrysymbolaccess` Get the value of the `symbolaccess` key for the entry with the given label:

```
\newcommand*{\glsentrysymbolaccess}[1]{%
  \@gls@entry@field{#1}{symbolaccess}%
}
```

`\glsentrysymbolpluralaccess` Get the value of the `symbolpluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentrysymbolpluralaccess}[1]{%
  \@gls@entry@field{#1}{symbolpluralaccess}%
}
```

`\glsentrydescaccess` Get the value of the `descriptionaccess` key for the entry with the given label:

```
\newcommand*{\glsentrydescaccess}[1]{%
  \@gls@entry@field{#1}{descaccess}%
}
```

`\glsentrydescpluralaccess` Get the value of the `descriptionpluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentrydescpluralaccess}[1]{%
  \@gls@entry@field{#1}{descpluralaccess}%
}
```

`\glsentryshortaccess` Get the value of the `shortaccess` key for the entry with the given label:

```
\newcommand*{\glsentryshortaccess}[1]{%
  \@gls@entry@field{#1}{shortaccess}%
}
```

`\glsentryshortpluralaccess` Get the value of the `shortpluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentryshortpluralaccess}[1]{%
  \@gls@entry@field{#1}{shortpluralaccess}%
}
```

`\glsentrylongaccess` Get the value of the `longaccess` key for the entry with the given label:

```

\newcommand*\glsentrylongaccess}[1]{%
  \@gls@entry@field{#1}{longaccess}%
}

```

`\glsentrylongpluralaccess` Get the value of the `longpluralaccess` key for the entry with the given label:

```

\newcommand*\glsentrylongpluralaccess}[1]{%
  \@gls@entry@field{#1}{longpluralaccess}%
}

```

`\glsentryuseriaccess` Get the value of the `user1access` key for the entry with the given label:

```

\newcommand*\glsentryuseriaccess}[1]{%
  \@gls@entry@field{#1}{useriaccess}%
}

```

`\glsentryuseriiaccess` Get the value of the `user2access` key for the entry with the given label:

```

\newcommand*\glsentryuseriiaccess}[1]{%
  \@gls@entry@field{#1}{useriiaccess}%
}

```

`\glsentryuseriiiaccess` Get the value of the `user3access` key for the entry with the given label:

```

\newcommand*\glsentryuseriiiaccess}[1]{%
  \@gls@entry@field{#1}{useriiiaccess}%
}

```

`\glsentryuserivaccess` Get the value of the `user4access` key for the entry with the given label:

```

\newcommand*\glsentryuserivaccess}[1]{%
  \@gls@entry@field{#1}{userivaccess}%
}

```

`\glsentryuservaccess` Get the value of the `user5access` key for the entry with the given label:

```

\newcommand*\glsentryuservaccess}[1]{%
  \@gls@entry@field{#1}{uservaccess}%
}

```

`\glsentryuserviaccess` Get the value of the `user6access` key for the entry with the given label:

```

\newcommand*\glsentryuserviaccess}[1]{%
  \@gls@entry@field{#1}{userviaccess}%
}

```

There are three types of replacement text:

Alt Description of some content that's non-textual (for example, an image). A word break is assumed after the content.

ActualText A character or sequence of characters that replaces textual content (for example, a dropped capital, a ligature or a symbol). No word break is assumed after the content.

E Expansion of an abbreviation to avoid ambiguity (for example, “St” could be short for “saint” or “street”).

Therefore, rather than having one command for all fields, it’s better to have a command dependent on the field type. For example, the `short` and `shortpl` keys would require `E`, the `symbol` key would require `ActualText`, and a field that contains an image would require `Alt`.

```
\glsfieldaccsupp{<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@glstdetoklabel{#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@glstdetoklabel{#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@glstdetoklabel{#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@glstdetoklabel{#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@glstdetoklabel{#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@glstdetoklabel{#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@glstdetoklabel{#2}@descaccess}%

```

```

        {#1}%
        {\@gls@fieldaccess@display{#2}{desc}{#1}{\glsentrydescaccess{#2}}}%
    }
descriptionpluralaccessdisplay As above but for the descriptionpluralaccess replacement text.
\newrobustcmd*{\glsdescriptionpluralaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#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@glstetoklabel{#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@glstetoklabel{#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@glstetoklabel{#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@glstetoklabel{#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@glstetoklabel{#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@glstetoklabel{#2}@useriiaccess}%

```

```

    {#1}%
    {\@gls@fieldaccess@display{#2}{userii}{#1}{\glstentryuseriiaccess{#2}}}%
  }

```

`\glsuseriiiaccessdisplay` As above but for the user3access replacement text.

```

\newrobustcmd*{\glsuseriiiaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#2}@useriiiaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useriii}{#1}{\glstentryuseriiiaccess{#2}}}%
}

```

`\glsuserivaccessdisplay` As above but for the user4access replacement text.

```

\newrobustcmd*{\glsuserivaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#2}@userivaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useriv}{#1}{\glstentryuserivaccess{#2}}}%
}

```

`\glsuservaccessdisplay` As above but for the user5access replacement text.

```

\newrobustcmd*{\glsuservaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#2}@uservaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{userv}{#1}{\glstentryuservaccess{#2}}}%
}

```

`\glsuserviaccessdisplay` As above but for the user6access replacement text.

```

\newrobustcmd*{\glsuserviaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#2}@userviaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{uservi}{#1}{\glstentryuserviaccess{#2}}}%
}

```

`\glsaccessdisplay` Gets the replacement text corresponding to the named key given by the first argument and calls the appropriate command defined above.

```

\newrobustcmd*{\glsaccessdisplay}[3]{%
  \ifcsundef{gls#1accessdisplay}%
  {%
    \PackageError{glossaries-accsupp}{No accessibility support
      for key ‘#1’}{%
    }%
  }%
  \csname gls#1accessdisplay\endcsname{#2}{#3}%
}

```

`\@gls@default@entryfmt` Deprecated `\@gls@default@entryfmt` was redefined to use accessibility information but has been removed in v4.50.

```

\glsentryfmt Redefine to use accessibility information.
\renewcommand*\glsentryfmt{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      Plural form
      \glscapscase
      {%
        Don't adjust case
        \ifglsused\glslabel
        {%
          Subsequent use
          \glspluralaccessdisplay
            {\glsentryplural{\glslabel}}{\glslabel}%
          \glsinsert
          }%
          {%
            First use
            \glsfirstpluralaccessdisplay
              {\glsentryfirstplural{\glslabel}}{\glslabel}%
            \glsinsert
            }%
            }%
            {%
              Make first letter upper case
              \ifglsused\glslabel
              {%
                Subsequent use.
                \glspluralaccessdisplay
                  {\Glsentryplural{\glslabel}}{\glslabel}%
                \glsinsert
                }%
                {%
                  First use
                  \glsfirstpluralaccessdisplay
                    {\Glsentryfirstplural{\glslabel}}{\glslabel}%
                  \glsinsert
                  }%
                  }%
                  {%
                    Make all upper case
                    \ifglsused\glslabel
                    {%

```

Subsequent use

```
\glsppluralaccessdisplay
  {\glssupercase{\glstentryplural{\glslabel}}}%
  {\glslabel}%
\glssupercase{\glsinsert}%
}%
{%
```

First use

```
\glsfirstpluralaccessdisplay
  {\glssupercase{\glstentryfirstplural{\glslabel}}}%
  {\glslabel}%
\glssupercase{\glsinsert}%
}%
}%
}%
{%
```

Singular form

```
\glscapscale
{%
```

Don't adjust case

```
\ifglsused\glslabel
{%
```

Subsequent use

```
\glstextaccessdisplay{\glstentrytext{\glslabel}}{\glslabel}%
\glsinsert
}%
{%
```

First use

```
\glsfirstaccessdisplay{\glstentryfirst{\glslabel}}{\glslabel}%
\glsinsert
}%
}%
{%
```

Make first letter upper case

```
\ifglsused\glslabel
{%
```

Subsequent use

```
\glstextaccessdisplay{\Glstentrytext{\glslabel}}{\glslabel}%
\glsinsert
}%
{%
```

First use

```
\glsfirstaccessdisplay{\Glstentryfirst{\glslabel}}{\glslabel}%
\glsinsert
}%
```

```
}%  
{%
```

Make all upper case

```
\ifglsused\glslabel  
{%
```

Subsequent use

```
\glstextaccessdisplay  
  {\glsuppercase{\glsentrytext{\glslabel}}}{\glslabel}%  
\glsuppercase{\glsinsert}%  
}%  
{%
```

First use

```
\glsfirstaccessdisplay  
  {\glsuppercase{\glsentryfirst{\glslabel}}}{\glslabel}%  
\glsuppercase{\glsinsert}%  
}%  
}%  
}%  
{%
```

Custom text provided in `\glsdisp`. (The insert should be empty at this point.)
The accessibility information, if required, will have to be explicitly included in
the custom text.

```
\glscustomtext\glsinsert  
}%  
}
```

`\glsngenacfmt` Redefine to include accessibility information.

```
\renewcommand*{\glsngenacfmt}{%  
  \ifdefempty\glscustomtext  
  {%  
    \ifglsused\glslabel  
    {%
```

Subsequent use:

```
\glsifplural  
{%
```

Subsequent plural form:

```
\glsupcase  
{%
```

Subsequent plural form, don't adjust case:

```
\acronymfont  
  {\glsshortpluralaccessdisplay  
   {\glsentryshortpl{\glslabel}}{\glslabel}}%  
\glsinsert  
}%  
{%
```

Subsequent plural form, make first letter upper case:

```
\acronymfont
{\glsshortpluralaccessdisplay
  {\Glsentryshortpl{\glslabel}}{\glslabel}}%
\glsinsert
}%
{%
```

Subsequent plural form, all caps:

```
\glsupercase
{\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:

```
\glsupercase
{\acronymfont{%
  \glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
  \glsinsert}%
}%
}%
{%
```

First use:

```
\glsifplural
{%
```

First use plural form:

```
\glscapscase
{%
```

First use plural form, don't adjust case:

```
\genplacrfullformat{\glslabel}{\glsinsert}%  
}%  
{%
```

First use plural form, make first letter upper case:

```
\Genplacrfullformat{\glslabel}{\glsinsert}%  
}%  
{%
```

First use plural form, all caps:

```
\glsuppercase  
{\genplacrfullformat{\glslabel}{\glsinsert}}%  
}%  
}%  
{%
```

First use singular form

```
\gls caps case  
{%
```

First use singular form, don't adjust case:

```
\genacrfullformat{\glslabel}{\glsinsert}%  
}%  
{%
```

First use singular form, make first letter upper case:

```
\Genacrfullformat{\glslabel}{\glsinsert}%  
}%  
{%
```

First use singular form, all caps:

```
\glsuppercase  
{\genacrfullformat{\glslabel}{\glsinsert}}%  
}%  
}%  
}%  
{%
```

User supplied text. (The insert should be empty at this point.) The accessibility information, if required, will have to be explicitly included in the custom text.

```
\gls custom text  
}%  
}
```

`\genacrfullformat` Redefine to include accessibility information.

```
\renewcommand*{\genacrfullformat}[2]{%  
  \gls long access display{\glsentrylong{#1}}{#1}#2\space  
  (\gls short access display{\protect\firstacronymfont{\glsentryshort{#1}}}{#1})%  
}
```



```

\Genacrfullformat Redefine to include accessibility information.
  \renewcommand*{\Genacrfullformat}[2]{%
    \glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space
    (\glsshortaccessdisplay{\protect\firstacronymfont{\Glsentryshort{#1}}}{#1})%
  }

\genplacrfullformat Redefine to include accessibility information.
  \renewcommand*{\genplacrfullformat}[2]{%
    \glslongpluralaccessdisplay{\glsentrylongpl{#1}}{#1}#2\space
    (\glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
  }

\Genplacrfullformat Redefine to include accessibility information.
  \renewcommand*{\Genplacrfullformat}[2]{%
    \glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}#2\space
    (\glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
  }

\@acrshort
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\glsentryshort{#2}}{#2}}#3%
    }%
  }

Call \@gls@link
  \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%

  \glspostlinkhook
}

\@Acrshort
\def\@Acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glsapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
    }%
  }

```

```

Call \@gls@link
  \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}

\@ACRshort
\def\@ACRshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax

    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay
        {\glsuppercase{\glsentryshort{#2}}}{#2}}#3%
    }%
  }%
Call \@gls@link
  \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}

\@acrlong
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax

    \let\glsifplural\@secondoftwo
    \let\glsapsase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{\glsentrylong{#2}}{#2}}#3%
    }%
  }%
Call \@gls@link
  \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}

\@Acrlong
\def\@Acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax

```



```

\@glsfirst@
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\glsentryfirst{#2}}{#2}#3}%
}

\@Glsfirst@
\def\@Glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\Glsentryfirst{#2}}{#2}#3}%
}

\@GLSfirst@
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsfirstaccessdisplay{\glsupercase{\glsentryfirst{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsplural@
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\glsentryplural{#2}}{#2}#3}%
}

\@Glsplural@
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\Glsentryplural{#2}}{#2}#3}%
}

\@GLSplural@
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glspluralaccessdisplay{\glsupercase{\glsentryplural{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsfirstplural@
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\glsentryfirstplural{#2}}{#2}#3}%
}

\@Glsfirstplural@
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\Glsentryfirstplural{#2}}{#2}#3}%
}

\@GLSfirstplural@
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsfirstpluralaccessdisplay{\glsupercase{\glsentryfirstplural{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

```

```

\@glsname@
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glsentryname{#2}}{#2}#3}%
}

\@Glsname@
\def\@Glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\Glsentryname{#2}}{#2}#3}%
}

\@GLSname@
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsnameaccessdisplay{\glsuppercase{\glsentryname{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

\@glsdesc@
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\glsentrydesc{#2}}{#2}#3}%
}

\@Glsdesc@
\def\@Glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
}

\@GLSdesc@
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsdescriptionaccessdisplay{\glsuppercase{\glsentrydesc{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

\@glsdescplural@
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\glsentrydescplural{#2}}{#2}#3}%
}

\@Glsdescplural@
\def\@Glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
}

\@GLSdescplural@
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsdescriptionpluralaccessdisplay{\glsuppercase{\glsentrydescplural{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

```

```

\@glssymbol@
\def\@glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
}

\@Glsymbol@
\def\@Glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}

\@GLSsymbol@
\def\@GLSsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glssymbolaccessdisplay{\glsupercase{\glsentrysymbol{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsymbolplural@
\def\@glsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsymbolpluralaccessdisplay{\glsentrysymbolplural{#2}}{#2}#3}%
}

\@Glsymbolplural@
\def\@Glsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsymbolpluralaccessdisplay{\Glsentrysymbolplural{#2}}{#2}#3}%
}

\@GLSsymbolplural@
\def\@GLSsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsymbolpluralaccessdisplay{\glsupercase{\glsentrysymbolplural{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsuseri@
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glsentryuseri{#2}}{#2}#3}%
}

\@Glsuseri@
\def\@Glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
}

\@GLSuseri@
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriaccessdisplay{\glsupercase{\glsentryuseri{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

```

```

\@glsuserii@
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glsentryuserii{#2}}{#2}#3}%
}

\@Glsuserii@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glsentryuserii{#2}}{#2}#3}%
}

\@GLSuserii@
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuseriiaccessdisplay{\glsuppercase{\glsentryuserii{#2}}}{#2}%
  \glsuppercase{#3}}%
}

\@glsuseriii@
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\glsentryuseriii{#2}}{#2}#3}%
}

\@Glsuseriii@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glsentryuseriii{#2}}{#2}#3}%
}

\@GLSuseriii@
\def\@GLSuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuseriiiaccessdisplay{\glsuppercase{\glsentryuseriii{#2}}}{#2}%
  \glsuppercase{#3}}%
}

\@glsuseriv@
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glsentryuseriv{#2}}{#2}#3}%
}

\@Glsuseriv@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
}

\@GLSuseriv@
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuserivaccessdisplay{\glsuppercase{\glsentryuseriv{#2}}}{#2}%
  \glsuppercase{#3}}%
}

```

```

\@glsuserv@
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\glsentryuserv{#2}}{#2}#3}%
}

\@Glsuserv@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\Glsentryuserv{#2}}{#2}#3}%
}

\@GLSuserv@
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuservaccessdisplay{\glsuppercase{\glsentryuserv{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

\@glsuservi@
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\glsentryuservi{#2}}{#2}#3}%
}

\@Glsuservi@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\Glsentryuservi{#2}}{#2}#3}%
}

\@GLSuservi@
\def\@GLSuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuserviaccessdisplay{\glsuppercase{\glsentryuservi{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

```

6.3 Displaying the Glossary

We need to redefine the way the glossary entries are formatted to include the accessibility support. The predefined glossary styles use `\glossentryname`, `\glossentrydesc` and `\glossentrysymbol`, but we need to provide compatibility with earlier versions in case users have defined their own styles using `\accsuppglossaryentryfield` and `\accsuppglossarysubentryfield`.

Now redefine `\glossentryname`, `\glossentrydesc` and `\glossentrysymbol` etc so they use the accessibility stuff.

```

\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\glsentryname{#1}}{#1}}%
  }%
}

```



```

\renewcommand*\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\Glsentryname{#1}}{#1}}%
  }%
}

\renewcommand*\glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}

\renewcommand*\Glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}

\renewcommand*\glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
  }%
}

\renewcommand*\Glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
  }%
}

```

\accsuppglossaryentryfield

```

\newcommand*\accsuppglossaryentryfield}[5]{%
  \glossaryentryfield{#1}%
  {\glsnameaccessdisplay{#2}{#1}}%
  {\glsdescriptionaccessdisplay{#3}{#1}}%
  {\glssymbolaccessdisplay{#4}{#1}}{#5}%
}

```

\accsuppglossarysubentryfield

```

\newcommand*\accsuppglossarysubentryfield}[6]{%
  \glossarysubentryfield{#1}{#2}%
  {\glsnameaccessdisplay{#3}{#2}}%
  {\glsdescriptionaccessdisplay{#4}{#2}}%
  {\glssymbolaccessdisplay{#5}{#2}}{#6}%
}

```

6.4 Acronyms

Redefine acronym styles provided by glossaries:

`long-short` $\langle long \rangle$ ($\langle short \rangle$) acronym style.

```
\renewacronymstyle{long-short}%
{%
```

Check for long form in case this is a mixed glossary.

```
\ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}##2\space
\glsshortaccessdisplay
{\protect\firstacronymfont{\glsentryshort{##1}}{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\glslongaccessdisplay{\Glsentrylong{##1}}{##1}##2\space
\glsshortaccessdisplay
{\protect\firstacronymfont{\glsentryshort{##1}}{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}##2\space
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glsentryshortpl{##1}}{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}##2\space
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glsentryshortpl{##1}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}}
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
```

`short-long` $\langle short \rangle$ ($\langle long \rangle$) acronym style.

```
\renewacronymstyle{short-long}%
{%
```

Check for long form in case this is a mixed glossary.

```
\ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
```

```

\glsshortaccessdisplay
  {\protect\firstacronymfont{\glentryshort{##1}}{##1}##2\space
(\glslongaccessdisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\glsshortaccessdisplay
  {\protect\firstacronymfont{\Glentryshort{##1}}{##1}##2\space
(\glslongaccessdisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
  {\protect\firstacronymfont{\glentryshortpl{##1}}{##1}##2\space
(\glslongpluralaccessdisplay
  {\glentrylongpl{##1}}{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
  {\protect\firstacronymfont{\Glentryshortpl{##1}}{##1}##2\space
(\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}

```

`long-short-desc` *⟨long⟩* (*{⟨short⟩}*) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{long-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}{##1}})%
}

```

`long-sc-short-desc` *⟨long⟩* (`\textsc{⟨short⟩}`) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{long-sc-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sc-short}%
}%

```

```

    {%
      \GlsUseAcrStyleDefs{long-sc-short}%
      \renewcommand*\GenericAcronymFields{}%
      \renewcommand*\acronymsort}[2]{##2}%
      \renewcommand*\acronymentry}[1]{%
        \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
      }
}

long-sm-short-desc <long> (\textsmaller{<short>}) acronym style that has an accompanying de-
scription (which the user needs to supply).
\renewacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
  }
}

short-long-desc <short> ({<long>}) acronym style that has an accompanying description (which
the user needs to supply).
\renewacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
  }
}

sc-short-long-desc <long> (\textsc{<short>}) acronym style that has an accompanying description
(which the user needs to supply).
\renewacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
}

```

```

\renewcommand*\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1))%
}

```

sm-short-long-desc *⟨long⟩* (`\textsmaller{⟨short⟩}`) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1))%
  }
}

```

dua *⟨long⟩* only acronym style.

```

\renewacronymstyle{dua}%
{%

```

Check for long form in case this is a mixed glossary.

```

\ifdefempty\glscustomtext
{%
  \ifglsahaslong{\glslabel}%
  {%
    \glsifplural
    {%

```

Plural form:

```

\gls caps case
{%

```

Plural form, don't adjust case:

```

\gls long plural access display {\glsentrylongpl{\glslabel}}{\glslabel}%
\gls insert
}%
{%

```

Plural form, make first letter upper case:

```

\gls long plural access display {\Glsentrylongpl{\glslabel}}{\glslabel}%
\gls insert
}%
{%

```

Plural form, all caps:

```

\gls long plural access display
{\glsupercase{\glsentrylongpl{\glslabel}}}{\glslabel}%

```

```

        \glsuppercase{\glsinsert}%
    }%
}%
{%-
Singular form
    \glscapscase
    {%-
Singular form, don't adjust case:
    \glslongaccessdisplay{\glsentrylong{\glslabel}}{\glslabel}\glsinsert
    }%
    {%-
Subsequent singular form, make first letter upper case:
    \glslongaccessdisplay{\Glsentrylong{\glslabel}}{\glslabel}\glsinsert
    }%
    {%-
Subsequent singular form, all caps:
    \glslongaccessdisplay
    {\glsuppercase
     {\glsentrylong{\glslabel}\glsinsert}}{\glslabel}%
    \glsuppercase{\glsinsert}%
    }%
}%
{%-
Not an acronym:
    \glsgenentryfmt
    }%
}%
{\glscustomtext\glsinsert}%
}%
{%-
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%-
    \glslink[##1]{##2}{%-
        \glslongaccessdisplay{\glsentrylong{##2}}{##2}##3\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}{##2})}}%
\renewcommand*{\Acrfullfmt}[3]{%-
    \glslink[##1]{##2}{%-
        \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%-
    \glslink[##1]{##2}{%-
        \glslongaccessdisplay
        {\glsuppercase{\glsentrylong{##2}}{##2}##3\space
         (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}{##2})}}%
\renewcommand*{\acrfullplfmt}[3]{%-
    \glslink[##1]{##2}{%-

```

```

\glslongpluralaccessdisplay
  {\glsentrylongpl{##2}}{##2}##3\space
(\glsshortpluralaccessdisplay
  {\acronymfont{\glsentryshortpl{##2}}{##2}})%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\Glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
      {\acronymfont{\glsentryshortpl{##2}}{##2}})%
  \renewcommand*{\ACRfullplfmt}[3]{%
    \glslink[##1]{##2}{%
      \glslongpluralaccessdisplay
        {\glsupercase{\glsentrylongpl{##2}}{##2}##3\space
      (\glsshortpluralaccessdisplay
        {\acronymfont{\glsentryshortpl{##2}}{##2}})%
    \renewcommand*{\glsentryfull}[1]{%
      \glslongaccessdisplay{\glsentrylong{##1}}\space
      (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
    }%
  \renewcommand*{\Glsentryfull}[1]{%
    \glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
  }%
  \renewcommand*{\glsentryfullpl}[1]{%
    \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}\space
    (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}{##1}})%
  }%
  \renewcommand*{\Glsentryfullpl}[1]{%
    \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space
    (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}{##1}})%
  }%
  \renewcommand*{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}}%
  \renewcommand*{\acronymsort}[2]{##1}%
  \renewcommand*{\acronymfont}[1]{##1}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}

```

dua-desc *<long>* only acronym style with user-supplied description.

```

\renewacronymstyle{dua-desc}%
{%
  \GlsUseAcrEntryDispStyle{dua}%
}%
{%
  \GlsUseAcrStyleDefs{dua}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\acronymfont{\glsentrylong{##1}}{##1}}%
\renewcommand*{\acronymsort}[2]{##2}%

```

}%

footnote *<short>*\footnote{*<long>*} acronym style.

```
\renewacronymstyle{footnote}%
```

{%

Check for long form in case this is a mixed glossary.

```
\ifglshaslong{\glslabel}{\glsгенacfmt}{\glsгенentryfmt}%
```

}%

{%

```
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
```

Need to ensure hyperlinks are switched off on first use:

```
\glshyperfirstfalse
```

```
\renewcommand*{\genacrfullformat}[2]{%
```

```
\glsshortaccessdisplay
```

```
{\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2%
```

```
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}}{##1}}%
```

}%

```
\renewcommand*{\Genacrfullformat}[2]{%
```

```
\glsshortaccessdisplay
```

```
{\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%
```

```
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}}{##1}}%
```

}%

```
\renewcommand*{\genplacrfullformat}[2]{%
```

```
\glsshortpluralaccessdisplay
```

```
{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2%
```

```
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%
```

}%

```
\renewcommand*{\Genplacrfullformat}[2]{%
```

```
\glsshortpluralaccessdisplay
```

```
{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%
```

```
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%
```

}%

```
\renewcommand*{\acronymentry}[1]{%
```

```
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
```

```
\renewcommand*{\acronymsort}[2]{##1}%
```

```
\renewcommand*{\acronymfont}[1]{##1}%
```

```
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
```

Don't use footnotes for \acrfull:

```
\renewcommand*{\acrfullfmt}[3]{%
```

```
\glslink[##1]{##2}{%
```

```
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}##3\space
```

```
(\glslongaccessdisplay{\glsentrylong{##2}}{##2}))}%
```

```
\renewcommand*{\Acrfullfmt}[3]{%
```

```
\glslink[##1]{##2}{%
```

```
\glsshortaccessdisplay{\acronymfont{\Glsentryshort{##2}}}{##2}##3\space
```

```
(\glslongaccessdisplay{\glsentrylong{##2}}{##2}))}%
```

```
\renewcommand*{\ACRfullfmt}[3]{%
```

```
\glslink[##1]{##2}{%
```



```

\glsshortaccessdisplay
  {\glsupercase
   {\acronymfont{\glsentryshort{##2}}{##2}##3\space
   (\glslongaccessdisplay{\glsentrylong{##2}}{##2})}}}%
\renewcommand*\acrfullplfmt}[3]{%
\glslink[##1]{##2}{%
\glsshortpluralaccessdisplay
  {\acronymfont{\glsentryshortpl{##2}}{##2}##3\space
  (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}}%
\renewcommand*\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{%
\glsshortpluralaccessdisplay
  {\acronymfont{\Glsentryshortpl{##2}}{##2}##3\space
  (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}}%
\renewcommand*\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%
\glsshortpluralaccessdisplay
  {\glsupercase
   {\acronymfont{\glsentryshortpl{##2}}{##2}##3\space
   (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}}}}%

```

Similarly for `\glsentryfull` etc:

```

\renewcommand*\glsentryfull}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}\space
(\glslongaccessdisplay{\glsentrylong{##1}}{##1})}%
\renewcommand*\Glsentryfull}[1]{%
\glsshortaccessdisplay{\acronymfont{\Glsentryshort{##1}}{##1}\space
(\glslongaccessdisplay{\glsentrylong{##1}}{##1})}%
\renewcommand*\glsentryfullpl}[1]{%
\glsshortpluralaccessdisplay
  {\acronymfont{\glsentryshortpl{##1}}{##1}\space
  (\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1})}%
\renewcommand*\Glsentryfullpl}[1]{%
\glsshortpluralaccessdisplay
  {\acronymfont{\Glsentryshortpl{##1}}{##1}\space
  (\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1})}%
}

```

`footnote-sc \textsc{<short>}\footnote{<long>}` acronym style.

```

\renewacronymstyle{footnote-sc}%
{%
\GlsUseAcrEntryDispStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}}
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*\acrpluralsuffix{\glsstextup{\glspluralsuffix}}%
}%

```

footnote-sm \textsmaller{<short>}\footnote{<long>} acronym style.

```
\renewacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}%
```

footnote-desc <short>\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
\renewacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
  }
}
```

footnote-sc-desc \textsc{<short>}\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
\renewacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
  }
}
```

footnote-sm-desc \textsmaller{<short>}\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
\renewacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}%
```

```
{%
\GlsUseAcrStyleDefs{footnote-sm}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}
```

```
\glsdefaultshortaccess{<long>}{<short>}
```

`\glsdefaultshortaccess`

Default shortaccess value.

```
\newcommand*\glsdefaultshortaccess}[2]{#1}
```

Use `\newacronymhook` to modify the key list to set the access text to the long version by default.

```
\renewcommand*\newacronymhook{%
\protected@edef\gls@keylist{%
shortaccess={\glsdefaultshortaccess{the\glslongtok}{the\glsshorttok}},%
shortpluralaccess={\glsdefaultshortaccess{the\glslongtok}{the\glsshorttok}},%
\the\glskeylisttok}%
\expandafter\glskeylisttok\expandafter{\@gls@keylist}%
}
```

`\DefaultNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\DescriptionFootnoteNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\DescriptionNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\FootnoteNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\SmallNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

The following are kept for compatibility with versions before 3.0:

`\glsshortaccesskey`

```
\newcommand*\glsshortaccesskey{\glsshortkey access}%
```

`\glsshortpluralaccesskey`

```
\newcommand*\glsshortpluralaccesskey{\glsshortpluralkey access}%
```

`\glslongaccesskey`

```
\newcommand*\glslongaccesskey{\glslongkey access}%
```

```
\glslongpluralaccesskey
    \newcommand*{\glslongpluralaccesskey}{\glslongpluralkey access}%
```

6.5 Debugging Commands

```
\showglongnameaccess
    \newcommand*{\showglongnameaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@access\endcsname
    }
```

```
\showglongtextaccess
    \newcommand*{\showglongtextaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@textaccess\endcsname
    }
```

```
\showglongpluralaccess
    \newcommand*{\showglongpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@pluralaccess\endcsname
    }
```

```
\showglongfirstaccess
    \newcommand*{\showglongfirstaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@firstaccess\endcsname
    }
```

```
\showglongfirstpluralaccess
    \newcommand*{\showglongfirstpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpluralaccess\endcsname
    }
```

```
\showglongsymbolaccess
    \newcommand*{\showglongsymbolaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolaccess\endcsname
    }
```

```
\showglongsymbolpluralaccess
    \newcommand*{\showglongsymbolpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolpluralaccess\endcsname
    }
```

```
\showglongdescaccess
    \newcommand*{\showglongdescaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@descaccess\endcsname
    }
```

```
\showglongdescpluralaccess
    \newcommand*{\showglongdescpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@descpluralaccess\endcsname
    }
```

`\showgloshortaccess`

```
\newcommand*{\showgloshortaccess}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@shortaccess\endcsname
}
```

`\showgloshortpluralaccess`

```
\newcommand*{\showgloshortpluralaccess}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@shortpluralaccess\endcsname
}
```

`\showglolongaccess`

```
\newcommand*{\showglolongaccess}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@longaccess\endcsname
}
```

`\showglolongpluralaccess`

```
\newcommand*{\showglolongpluralaccess}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@longpluralaccess\endcsname
}
```

7 Backwards Compatibility

7.1 glossaries-compatible-207

Provides compatibility with version 2.07 and below. This uses original glossaries indy and makeindex formatting, so can be used with old documents that had customized style files, but hyperlinks may not work properly.

Now only available with rollback.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossaries-compatible-207-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-compatible-207-2021-11-01.sty}
\DeclareCurrentRelease{v4.53}{2023-09-29}
```

Identify package.

```
\ProvidesPackage{glossaries-compatible-207}[2023/09/29 v4.53 (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{v4.53}{2023-09-29}
```

Identify package.

```
\ProvidesPackage{glossaries-compatible-307}[2023/09/29 v4.53 (NLCT)]
```

Trigger error and that's it.

```
\PackageError{glossaries}
{glossaries-compatible-307.sty no longer available. Use rollback if you need it}%
{Try \string\usepackage{glossaries} [=v4.46]}
```

8 Rollback

8.1 Rollback v4.46 (glossaries-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{ifthen}
\RequirePackage{xkeyval}[2006/11/18]
\RequirePackage{mfirstuc}
\RequirePackage{textcase}
\renewcommand*{\mfirstucMakeUppercase}{\MakeTextUppercase}%
\RequirePackage{xfor}
\RequirePackage{datatool-base}
\RequirePackage{amsgen}
\RequirePackage{etoolbox}
\newif\if@gls@docloaded
\@ifpackageloaded{doc}%
{
  \@gls@docloadedtrue
}%
{
  \@gls@docloadedfalse
}
\if@gls@docloaded
  \let\glsorg@theglossary\theglossary
  \let\glsorg@endtheglossary\endtheglossary
  \let\glsorg@PrintChanges\PrintChanges
  \renewcommand{\PrintChanges}{%
    \begingroup
      \let\theglossary\glsorg@theglossary
      \let\endtheglossary\glsorg@endtheglossary
      \glsorg@PrintChanges
    \endgroup
  }
\fi
```

```

\newif\if@gls@debug
\define@choicekey{glossaries.sty}{debug}[\gls@debug@val\gls@debug@nr]{%
{true,false,showtargets,showaccsupp}[true]{%
\ifcase\gls@debug@nr\relax
% debug=true
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\let\@glsshowtarget\@gobble
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\or
% debug=false
\@gls@debugfalse
\let\@glsshowtarget\@gobble
\let\@glsshowaccsupp\@gobblethree
\PackageInfo{glossaries}{debug mode OFF}%
\or
% debug=showtargets
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowtarget}{\@glsshowtarget}%
\or
% debug=showaccsupp
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
\fi
}
\newcommand*\glsshowtarget}[1]{%
\ifmmode
\nfss@text{\glsshowtargetfont [#1]}%
\else
\ifinner
{\glsshowtargetfont [#1]}%

```

```

\else
  \glsshowtargetouter{#1}%
\fi
\fi
}
\newcommand*\glsshowtargetouter}[1]{%
  \glsshowtargetsymbolsymbol\marginpar{\glsshowtargetsymbolsymbol\glsshowtargetfont #1}}
\newcommand*\glsshowtargetsymbolsymbol{{\tiny$\triangleright$}}
\newcommand*\glsshowtargetfont{\ttfamily\footnotesize}
\newcommand*\@glsshowtarget}[1]{%
\newrobustcmd*\@@glsshowtarget}[1]{%
  \begingroup
  \protected@edef\@gls@tmp{#1}%
  \@onelevel@sanitize\@gls@tmp
  \expandafter\glsshowtarget\expandafter{\@gls@tmp}%
  \endgroup
}
\newcommand*\@glsshowaccsupp}[3]{%
\newrobustcmd*\glsshowaccsupp}[3]{%
  \ifstrempy{#1}%
  {\@@glsshowtarget{/#2 (#3)}}%
  {\@@glsshowtarget{/#2 (#3) [#1]}}%
}
\newcommand*\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries\space (or move
  \string\newglossaryentry\space
  definitions into the preamble)}}%
{You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key. It may
  be that the 'see' key has been written to the .glsdefs
  file from the previous run, in which case you need to
  move your definitions
  to the preamble if you don't want to use
  \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoinde}x{%
  [\gls@seenoinde@val\gls@seenoinde@nr]{error,warn,ignore}{%
  \ifcase\gls@seenoinde@nr
  \renewcommand*\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries}%
  {You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key}%
  }%
}

```



```

\or
  \renewcommand*\@gls@see@noindex}{%
    \GlossariesWarning{'\gls@xr@key' key ignored}%
  }%
\or
  \renewcommand*\@gls@see@noindex}{}%
\fi
}
\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}%
  {\newcommand*\@@glossarysec}{section}}%
  {\newcommand*\@@glossarysec}{chapter}}
\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
subsection,subsection,paragraph,subparagraph}[section]{%
  \renewcommand*\@@glossarysec{#1}}
\newcommand*\@@glossarysecstar}{*}%
\newcommand*\@@glossaryseclabel}{%
\newcommand*\glsautoprefix{}%
\define@choicekey{glossaries.sty}{numberedsection}%
  [\gls@numberedsection@val\gls@numberedsection@nr]{%
false,nolabel,autolabel,nameref}[nolabel]{%
  \ifcase\gls@numberedsection@nr\relax
    \renewcommand*\@@glossarysecstar}{*}%
    \renewcommand*\@@glossaryseclabel}{%
  \or
    \renewcommand*\@@glossarysecstar}{}%
    \renewcommand*\@@glossaryseclabel}{%
  \or
    \renewcommand*\@@glossarysecstar}{}%
    \renewcommand*\@@glossaryseclabel}{%
    \label{\glsautoprefix\@glo@type}}%
  \or
    \renewcommand*\@@glossarysecstar}{*}%
    \renewcommand*\@@glossaryseclabel}{%
    \protected@edef\@currentlabelname{\glossarytoctitle}%
    \label{\glsautoprefix\@glo@type}}%
  \fi
}
\@ifpackageloaded{classicthesis}
{\newcommand*\@glossary@default@style}{index}}
{\newcommand*\@glossary@default@style}{list}}
\define@key{glossaries.sty}{style}{%
  \def\@glossary@default@style{#1}%
}
\newcommand*\@gls@declareoption}[2]{%
  \DeclareOptionX{#1}{#2}%
  \DeclareOption{#1}{#2}%
}
\newcommand*\glossaryentrynumbers}[1]{#1\gls@save@numberlist{#1}}

```

```

\@gls@declareoption{nonumberlist}{%
  \renewcommand*\glossaryentrynumbers}[1]{\gls@save@numberlist{#1}}%
}
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{%
\gls@savenumberlistfalse
\newcommand*\glo@seeautonumberlist{}
\@gls@declareoption{seeautonumberlist}{%
  \renewcommand*\glo@seeautonumberlist}{%
    \def\glo@prefix{\glsnextpages}%
  }%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{%
\glsesclocationstrue
\newcommand*\gls@loadlong{\RequirePackage{glossary-long}[=v4.46]}
\@gls@declareoption{nolong}{\renewcommand*\gls@loadlong{}}
\IfFileExists{supertabular.sty}{%
  \newcommand*\gls@loadsuper{\RequirePackage{glossary-super}[=v4.46]}%
  \newcommand*\gls@loadsuper{}}
\@gls@declareoption{nosuper}{\renewcommand*\gls@loadsuper{}}
\newcommand*\gls@loadlist{\RequirePackage{glossary-list}[=v4.46]}
\@gls@declareoption{nolist}{%
  \renewcommand*\gls@loadlist}{%
    \ifdefstring{\glossary@default@style}{list}%
    {\let\glossary@default@style\relax}%
  }%
}
\newcommand*\gls@loadtree{\RequirePackage{glossary-tree}[=v4.46]}
\@gls@declareoption{notree}{\renewcommand*\gls@loadtree{}}
\@gls@declareoption{nostyles}{%
  \renewcommand*\gls@loadlong}{%
  \renewcommand*\gls@loadsuper}{%
  \renewcommand*\gls@loadlist}{%
  \renewcommand*\gls@loadtree}{%
  \let\glossary@default@style\relax
}
\newcommand*\glspostdescription}{%
  \ifglsnopostdot\else.\spacefactor\sfcode'\. \fi
}
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{%
\glsnopostdotfalse
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{%
\glsnogroupskipfalse
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{%
\@ifclassloaded{memoir}
{%
  \glsucmarktrue
}%
}
}
\glsucmarkfalse

```

```

}
\newcommand*{\@gls@define@glossaryentrycounter}{%
  \ifglentrycounter
  \ifundef\c@glossaryentry
  {%
    \ifx\@gls@counterwithin\@empty
    \newcounter{glossaryentry}%
    \else
    \newcounter{glossaryentry}[\@gls@counterwithin]%
    \fi
    \def\theH@glossaryentry{\currentglossary.\theglossaryentry}%
  }%
  }%
\fi
}
\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\glentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
  \renewcommand*{\@gls@counterwithin}{#1}%
  \glentrycountertrue
  \@gls@define@glossaryentrycounter
}
\newcommand*{\@gls@counterwithin}{}
\newcommand{\@gls@define@glossarysubentrycounter}{%
  \ifundef\c@glossarysubentry
  {%
    \ifglssubentrycounter
    \ifglentrycounter
    \newcounter{glossarysubentry}[glossaryentry]%
    \else
    \newcounter{glossarysubentry}%
    \fi
    \def\theH@glossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
  \fi
  }%
  }%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
\glssubentrycounterfalse
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none}{%
  \renewcommand*{\@glo@default@sorttype}{#1}%
  \csname @gls@setupsort@#1\endcsname
}
\newcommand*{\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\gls@sortnumberfmt
{\csname glossary@\@glo@type @sortcount\endcsname}}%
\@glo@storeentry{##1}%
\fi
}%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*{\@gls@setupsort@none}{%
\def\do@glo@storeentry##1{%
\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
\expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort
}%
\def\@gls@setsort##1{%
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
{Option sort=none not allowed with \string##1}%
{(Use sort=def instead)}}%
}
\newcommand*{\glsdefmain}{%
\if@gls@docloaded
\newglossary[glg2]{main}{gls2}{glo2}{\glossaryname}%
\else
\newglossary{main}{gls}{glo}{\glossaryname}%
\fi
\newcommand*{\gls@tr@set@main@toctitle}{%
\translatelet{\glossarytoctitle}{Glossary}%
}%
}
\newcommand*{\glsdefaulttype}{main}
\newcommand*{\acronymtype}{\glsdefaulttype}
\@gls@declareoption{nomain}{%
\let\glsdefaulttype\relax
\renewcommand*{\glsdefmain}{}%
}
\define@boolkey{glossaries.sty}{gls}{acronym}[true]{%
\ifglsacronym
\renewcommand{\@gls@do@acronymsdef}{%
\DeclareAcronymList{acronym}%
\newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
}

```

```

        \renewcommand*\acronymtype{acronym}%
        \newcommand*\gls@tr@set@acronym@toctitle}{%
            \translatelet{\glossarytoctitle}{Acronyms}%
        }%
    }%
\else
    \let\gls@do@acronymsdef\relax
\fi
}
\AtBeginDocument{%
    \ifglsacronym
        \ifbool{glscompatible-3.07}{%
            }{%
                \providecommand*\printacronyms[1][1]{%
                    \printglossary[type=\acronymtype,#1]}%
            }%
        \fi
    }
\newcommand*\gls@do@acronymsdef{}
\gls@declareoption{acronyms}{%
    \glsacronymtrue
    \def\gls@do@acronymsdef{%
        \DeclareAcronymList{acronym}%
        \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
        \renewcommand*\acronymtype{acronym}%
        \newcommand*\gls@tr@set@acronym@toctitle}{%
            \translatelet{\glossarytoctitle}{Acronyms}%
        }%
    }%
}
\newcommand*\glsacronymlists{}
\newcommand*\@addtoacronymlists[1]{%
    \ifx\glsacronymlists\@empty
        \protected@xdef\glsacronymlists{#1}%
    \else
        \protected@xdef\glsacronymlists{\glsacronymlists,#1}%
    \fi
}
\newcommand*\DeclareAcronymList[1]{%
    \glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}
\newcommand*\glsIfListOfAcronyms[1]{%
    \edef\@do@gls@islistofacronyms{%
        \noexpand\gls@islistofacronyms{#1}{\glsacronymlists}}%
    \@do@gls@islistofacronyms
}
\newcommand*\@gls@islistofacronyms[4]{%
    \def\gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%
        \def\@gls@before{##1}\def\@gls@after{##2}}%
}

```

```

\gls@islistofacronyms,#2,#1,\@nil\end@gls@islistofacronyms
\ifx\@gls@after\@nnil
  #4%
\else
  #3%
\fi
}
\newif\if@gls@isacronymlist
\newcommand*\@gls@checkisacronymlist}[1]{%
  \glsIfListOfAcronyms{#1}%
  {\@gls@isacronymlisttrue}{\@gls@isacronymlistfalse}%
}
\newcommand*\@SetAcronymLists}[1]{%
  \renewcommand*\@gls@acronymlists{#1}%
}
\define@key{glossaries.sty}{acronymlists}{%
  \DeclareAcronymList{#1}%
}
\newcommand*\@gls@counter{page}
\define@key{glossaries.sty}{counter}{%
  \renewcommand*\@gls@counter{#1}%
}
\newcommand*\@gls@nohyperlist{}
\newcommand*\@GlsDeclareNoHyperList}[1]{%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*\@gls@nohyperlist{#1}%
  }%
  {%
    \appto\@gls@nohyperlist{,#1}%
  }%
}
\define@key{glossaries.sty}{nohypertypes}{%
  \GlsDeclareNoHyperList{#1}%
}
\newcommand*\@GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{#1}%
}
\newcommand*\@GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{#1}%
}
\newcommand*\@glosortentrieswarning{%
  \typeout{Using TeX to sort glossary entries---this may
  take a while}%
}
\@gls@declareoption{nowarn}{%
  \if@gls@debug
    \GlossariesWarning{Warnings can't be suppressed in debug mode}%
  \else
    \renewcommand*\@GlossariesWarning}[1]{%

```

```

        \renewcommand*\GlossariesWarningNoLine}[1]{}%
        \renewcommand*\glosortentrieswarning}{}%
        \renewcommand*\@gls@missinglang@warn}[2]{}%
    \fi
}
\newcommand*\@gls@missinglang@warn}[2]{%
    \PackageWarningNoLine{glossaries}%
    {No language module detected for ‘#1’.\MessageBreak
    Language modules need to be installed separately.\MessageBreak
    Please check on CTAN for a bundle called\MessageBreak
    ‘glossaries-#2’ or similar}%
}
\@gls@declareoption{noLANGwarn}{%
    \renewcommand*\@gls@missinglang@warn}[2]{}%
}
\newcommand*\@gls@warnonglossdefined}{%
    \GlossariesWarning{Overriding \string\printglossary}%
}
\newcommand*\@gls@warnontheglossdefined}{%
    \GlossariesWarning{Overriding ‘theglossary’ environment}%
}
\@gls@declareoption{noRedefwarn}{%
    \renewcommand*\@gls@warnonglossdefined}{}%
    \renewcommand*\@gls@warnontheglossdefined}{}%
}
\newcommand*\@gls@sanitizedesc}{%
}
\newcommand*\glssetexpandfield}[1]{%
    \csdef{gls@assign@#1@field}##1##2{%
        \@gls@expand@field{##1}{#1}{##2}%
    }%
}
\newcommand*\glssetnoexpandfield}[1]{%
    \csdef{gls@assign@#1@field}##1##2{%
        \@gls@noexpand@field{##1}{#1}{##2}%
    }%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}
\glssetnoexpandfield{desclplural}
\newcommand*\@gls@sanitizename}{%
\glssetnoexpandfield{name}
\newcommand*\@gls@sanitizesymbol}{%
\glssetnoexpandfield{symbol}
\glssetnoexpandfield{symbolplural}
\newcommand*\@gls@sanitizesort}{%
    \ifgls@sanitizesort
        \@gls@sanitizesort
    \else
        \@gls@nosanitizesort
    \fi
}

```



```

\fi
}
\newcommand*{\@@gls@sanitizesort{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@@gls@nosanitizesort}{%}
\newcommand*{\@gls@noidx@sanitizesort{%
  \ifdefvoid\@glo@sort
  }{%
  {%
  \expandafter\@@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
  \def\@glo@sort{#1#2}%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@@gls@noidx@nosanitizesort}{%
  \ifdefvoid\@glo@sort
  }{%
  {%
  \expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
  \bgroup
  \glsnoidxstripaccents
  \protected@xdef\@@glo@sort{#1#2}%
  \egroup
  \let\@glo@sort\@@glo@sort
}
\newcommand*{\glsnoidxstripaccents{%
  \let\IeC\@firstofone
  \let\add@accent@\@secondoftwo
  \let\@text@composite@x@\@secondoftwo
  \let\@tabacckludge@\@secondoftwo
  \expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
  \let'\@firstofone
  \let'\@firstofone
  \let^\@firstofone
  \let"@@firstofone
  \let\u\@firstofone
  \let\t\@firstofone
  \let\d\@firstofone
  \let\r\@firstofone
  \let=\@firstofone
  \let.\@firstofone

```

```

\let~\@firstofone
\let\v\@firstofone
\let\H\@firstofone
\let\c\@firstofone
\let\b\@firstofone
\let\a\@secondoftwo
\def\AE{AE}%
\def\ae{ae}%
\def\OE{OE}%
\def\oe{oe}%
\def\AA{AA}%
\def\aa{aa}%
\def\L{L}%
\def\l{l}%
\def\O{O}%
\def\o{o}%
\def\SS{SS}%
\def\ss{ss}%
\def\th{th}%
\def\TH{TH}%
\def\dh{dh}%
\def\DH{DH}%
}
\@ifl@t@r\fmtversion{2019/10/01}
{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}
\define@boolkey[glS]{sanitize}{description}[true]{%
\GlossariesWarning{sanitize={description} package option deprecated}%
\ifglS@sanitize@description
\glssetnoexpandfield{desc}%
\glssetnoexpandfield{descplural}%
\else
\glssetexpandfield{desc}%
\glssetexpandfield{descplural}%
\fi
}
\define@boolkey[glS]{sanitize}{name}[true]{%
\GlossariesWarning{sanitize={name} package option deprecated}%
\ifglS@sanitize@name
\glssetnoexpandfield{name}%
\else
\glssetexpandfield{name}%
\fi
}
\define@boolkey[glS]{sanitize}{symbol}[true]{%
\GlossariesWarning{sanitize={symbol} package option deprecated}%
\ifglS@sanitize@symbol
\glssetnoexpandfield{symbol}%

```

```

        \glsssetnoexpandfield{symbolplural}%
    \else
        \glsssetexpandfield{symbol}%
        \glsssetexpandfield{symbolplural}%
    \fi
}
\define@boolkey{glossaries.sty}[gls]{sanitizesort}[true]{%
    \ifglsssanitizesort
        \glsssetnoexpandfield{sortvalue}%
        \renewcommand*{\@gls@noidx@setsanitizesort}{%
            \glsssanitizesorttrue
            \glsssetnoexpandfield{sortvalue}%
        }%
    \else
        \glsssetexpandfield{sortvalue}%
        \renewcommand*{\@gls@noidx@setsanitizesort}{%
            \glsssanitizesortfalse
            \glsssetexpandfield{sortvalue}%
        }%
    \fi
}
\glsssanitizesorttrue
\glsssetnoexpandfield{sortvalue}%
\newcommand*{\@gls@noidx@setsanitizesort}{%
    \glsssanitizesortfalse
    \glsssetexpandfield{sortvalue}%
}
\define@choicekey[gls]{sanitize}{sort}{true,false}[true]{%
    \setbool{glsssanitizesort}{#1}%
    \ifglsssanitizesort
        \glsssetnoexpandfield{sortvalue}%
    \else
        \glsssetexpandfield{sortvalue}%
    \fi
}
\GlossariesWarning{sanitize={sort} package option
    deprecated. Use sanitizesort instead}%
}
\define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
    \ifthenelse{\equal{#1}{none}}{%
        {%
            \GlossariesWarning{sanitize package option deprecated}%
            \glsssetexpandfield{name}%
            \glsssetexpandfield{symbol}%
            \glsssetexpandfield{symbolplural}%
            \glsssetexpandfield{desc}%
            \glsssetexpandfield{descplural}%
        }%
    }%
    \setkeys[gls]{sanitize}{#1}%
}%

```

```

}
\newif\ifglstranslate
\newcommand*\@gls@usetranslator{%
  \@ifpackageloaded{polyglossia}%
  {%
    \let\glsifusetranslator\@secondoftwo
  }%
  {%
    \@ifpackageloaded{babel}%
    {%
      \IfFileExists{translator.sty}%
      {%
        \RequirePackage{translator}%
        \let\glsifusetranslator\@firstoftwo
      }%
    }%
  }%
  {}%
}%
}
\newcommand{\glsifusedtranslatordict}[3]{%
  \glsifusetranslator
  {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
  {#3}%
}
\@gls@declareoption{nottranslate}{%
  \glstranslatefalse
  \let\@gls@usetranslator\relax
  \let\glsifusetranslator\@secondoftwo
}
\define@choicekey{glossaries.sty}{translate}%
  [\gls@translate@val\gls@translate@nr]%
  {true,false,babel}[true]%
  {%
    \ifcase\gls@translate@nr\relax
    \glstranslatetrue
    \renewcommand*\@gls@usetranslator{%
      \@ifpackageloaded{polyglossia}%
      {%
        \let\glsifusetranslator\@secondoftwo
      }%
    }%
    \@ifpackageloaded{babel}%
    {%
      \IfFileExists{translator.sty}%
      {%
        \RequirePackage{translator}%
        \let\glsifusetranslator\@firstoftwo
      }%
    }%
  }%
  {}%
}

```

```

        }%
        {}%
    }%
}
\or
\glstranslatefalse
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\or
\glstranslatetrue
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\fi
}
\glstranslatefalse
\let\glsifusetranslator\@secondoftwo
\@ifpackageloaded{translator}%
{%
\glstranslatetrue
\let\glsifusetranslator\@firstoftwo
}%
{%
\@for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
{
\@ifpackageloaded{\gls@thissty}%
{%
\glstranslatetrue
\@endfortrue
}%
{}%
}
}
\define@boolkey{glossaries.sty}[gls]{indexonlyfirst}[true]{}
\glsindexonlyfirstfalse
\define@boolkey{glossaries.sty}[gls]{hyperfirst}[true]{}
\glshyperfirsttrue
\newcommand*{\@gls@setacrstyle}{}
\define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{%
\ifbool{glsacrdescription}%
{}%
{%
\renewcommand*{\@gls@sanitizedesc}{}}%
}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{description}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{%

```

```

\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{dua}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{shortcuts}[true]{}
\newcommand*{\glsorder}{word}
\newcommand*{\@glsorder}[1]{}
\define@choicekey{glossaries.sty}{order}{word,letter}{%
\def\glsorder{#1}}
\newif\ifglsxindy
\glsxindyfalse
\@gls@declareoption{makeindex}{\glsxindyfalse}
\define@boolkey[gls]{xindy}{glsnumbers}[true]{}
\glsxindy@glsnumberstrue
\def\@xdy@main@language{\languagename}%
\define@key[gls]{xindy}{language}{\def\@xdy@main@language{#1}}
\ifcsundef{inputencodingname}{%
\def\gls@codepage{}}%
\def\gls@codepage{\inputencodingname}
}
\define@key[gls]{xindy}{codepage}{\def\gls@codepage{#1}}
\define@key{glossaries.sty}{xindy}[]{}
\glsxindytrue
\setkeys[gls]{xindy}{#1}%
}
\@gls@declareoption{xindygloss}{%
\glsxindytrue
}
}
\@gls@declareoption{xindynoglsnumbers}{%
\glsxindytrue
\glsxindy@glsnumbersfalse
}
\providecommand{\@domakeglossaries}[1]{#1}
\@gls@declareoption{disablemakegloss}{%
\ifdefequal\makeglossaries\@no@makeglossaries
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect
(\string\makenoidxglossaries\space has already been used)}}%
}%
{%
\ifdefequal\@makeglossary\@gobble
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect

```

```

        (\string\makeglossaries\space has already been used)}%
    }%
    {%
    \renewcommand{\@domakeglossaries}[1]{%
        \PackageInfo{glossaries}{\string\makeglossaries\space and
        \string\makenoidxglossaries\space have been disabled}%
    }%
    }%
    }%
}
\@gls@declareoption{restoremakegloss}{%
    \ifdefequal\makeglossaries\@no@makeglossaries
    {%
        \GlossariesWarning{Option ‘restoremakegloss’ has no effect
        (\string\makenoidxglossaries\space has already been used)}%
    }%
    {%
        \ifdefequal\@makeglossary\@gobble
        {%
            \GlossariesWarning{Option ‘restoremakegloss’ has no effect
            (\string\makeglossaries\space has already been used)}%
        }%
        {%
            \PackageInfo{glossaries}{\string\makeglossaries\space and
            \string\makenoidxglossaries\space have been restored}%
            \let\@domakeglossaries\@firstofone
        }%
    }%
}
}
\newcommand*\@do@write@glslabels{%
    \AtEndDocument{\@do@write@glslabels}%
    \let\@do@write@glslabels\relax
}
\newcommand*\@do@write@glslabels{%
    \newwrite\@gls@labelsfile
    \immediate\openout\@gls@labelsfile=\jobname.glslabels
    \forallglsentries[\@glo@types,\@ignored@glossaries]{\@glsentry}%
    {\ifdefempty{\@glsentry}{\immediate\write\@gls@labelsfile{\@glsentry}}}%
    \immediate\closeout\@gls@labelsfile
}
\@gls@declareoption{writeglslabels}{\@do@write@glslabels}
\newif\ifglsautomake
\newcommand{\gls@automake@nr}{1}
\define@choicekey{glossaries.sty}{automake}%
    [\gls@automake@val\gls@automake@nr]{true,false,immediate}[true]{%
    \ifnum\gls@automake@nr=1\relax
        \glsautomakefalse
    \else
        \glsautomaketrue
    \fi
}

```

```

\ifglsautomake
  \renewcommand*{\@gls@doautomake}{%
    \PackageError{glossaries}{You must use
    \string\makeglossaries\space with automake=true}
    {%
      Either remove the automake=true setting or
      add \string\makeglossaries\space to your document preamble.%
    }%
  }%
\else
  \renewcommand*{\@gls@doautomake}{}%
\fi
}
\glsautomakefalse
\newcommand*{\@gls@doautomake}{}
\AtEndDocument{\@gls@doautomake}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
  \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@ption
      \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
    \fi
  \fi
}
\ProcessOptionsX
\RequirePackage{glossaries-compatible-307}[=v4.46]
\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,notranslate,nolong,nosuper,notree,nostyles,%
nomain,noglossaryindex}
\newcommand*{\setupglossaries}[1]{%
  \renewcommand*{\@gls@setacrstyle}{}%
  \ifglsacrshortcuts
    \def\@gls@setupshortcuts{\glsacrshortcutstrue}%
  \else
    \def\@gls@setupshortcuts{%
      \ifglsacrshortcuts

```



```

\expandafter\@disable@premakecs\@thiscs%
}}
\newcommand*\@disable@premakecs}[1]{%
  \def#1{\PackageError{glossaries}{\string#1\space may only be
    used before \string\makeglossaries}{You can't use
    \string#1\space after \string\makeglossaries}}%
}
\providecommand*\glossaryname{Glossary}
\providecommand*\acronymname{Acronyms}
\newcommand*\glssettoctitle}[1]{%
  \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}
\providecommand*\entryname{Notation}
\providecommand*\descriptionname{Description}
\providecommand*\symbolname{Symbol}
\providecommand*\pagelistname{Page List}
\providecommand*\glsymbolsgroupname{Symbols}
\providecommand*\glsnumbersgroupname{Numbers}
\newcommand*\glspluralsuffix}{s}
\newcommand*\glsacrpluralsuffix}{\glspluralsuffix}
\newcommand*\glsupacrpluralsuffix}{\glstextup{\glsacrpluralsuffix}}
\providecommand*\seename}{see}
\providecommand*\andname}{\&}
\newcommand*\RequireGlossariesLang}[1]{%
  \ifundefined{ver@glossaries-#1.1df}{\input{glossaries-#1.1df}}{}}%
}
\newcommand*\ProvidesGlossariesLang}[1]{%
  \ProvidesFile{glossaries-#1.1df}%
}
\newcommand*\addglossarytocaptions}[1]{}
\ifglstranslate
  \RequirePackage{tracklang}
  \@gls@usetranslator
  \@ifpackageloaded{translator}
  {%
    \ifboolexpr
    {
      test {\ifdefstring{\trans@languages}{English}}
      and not
      test {\ifdefstring{bbl@loaded}{english}}
    }
    {%
      \let\glsifusetranslator\@secondoftwo
    }%
    {%
      \usedictionary{glossaries-dictionary}%
      \renewcommand*\addglossarytocaptions}[1]{%
        \ifcsundef{captions#1}{}%
        {%
          \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
          \expandafter\toks@\expandafter{\@gls@tmp

```

```

        \renewcommand*{\glossaryname}{\translate{Glossary}}%
    }%
    \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
  }%
} %
} %
{} %
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
  }%
} %
{} %
\glsifusetranslator
{%
  \renewcommand*{\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*{\glossaryname}{\translate{Glossary}}%
  \renewcommand*{\acronymname}{\translate{Acronyms}}%
  \renewcommand*{\entryname}{\translate{Notation (glossaries)}}%
  \renewcommand*{\descriptionname}{%
    \translate{Description (glossaries)}}%
  \renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}%
  \renewcommand*{\pagelistname}{%
    \translate{Page List (glossaries)}}%
  \renewcommand*{\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*\glstfilebase{#1}%
  \ifglxindy
    \def\istfilename{\glstfilebase.xdy}
  \else
    \def\istfilename{\glstfilebase.ist}
  \fi
}
\@onlypremakeg\setStyleFile
\ifglxindy
  \def\istfilename{\glstfilebase.xdy}
\else
  \def\istfilename{\glstfilebase.ist}
\fi
\newcommand*\glstfilebase{\jobname}
\newcommand*\@istfilename}[1]{
\newcommand*\glscpositor}{.}
\newcommand*\glstSetCompositor}[1]{%
  \renewcommand*\glscpositor{#1}}
\@onlypremakeg\glstSetCompositor
\newcommand*\@glstAlphacompositor{\glscpositor}
\ifglxindy
  \newcommand*\glstSetAlphaCompositor[1]{%
    \renewcommand*\@glstAlphacompositor{#1}}
\else
  \newcommand*\glstSetAlphaCompositor[1]{%
    \glstnoxywarning\glstSetAlphaCompositor}
\fi
\@onlypremakeg\glstSetAlphaCompositor
\newcommand*\glstsuffixF{}
\newcommand*\glstSetSuffixF}[1]{%
  \renewcommand*\glstsuffixF{#1}}
\@onlypremakeg\glstSetSuffixF
\newcommand*\glstsuffixFF{}
\newcommand*\glstSetSuffixFF}[1]{%
  \renewcommand*\glstsuffixFF{#1}}
}
\ifcsundef{hyperlink}%
{%
  \newcommand*\glstnumberformat}[1]{#1}%
}%
{%
  \newcommand*\glstnumberformat}[1]{\glsthypernumber{#1}}%
}

```

```

\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*{\glossarypreamble}{%
  \csuse{@glossarypreamble@\currentglossary}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaultttype]{%
  \ifglossaryexists*{#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*{\glossarypostamble}{}
\newcommand*{\glossarysection}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@p@glossarysection{#1}{#2}%
  }%
  \gls@glossarymark{\glossarytoctitle}%
}
\ifcsundef{glossarymark}%
{%
  \newcommand{\gls@glossarymark}[1]{\glossarymark{#1}}
}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand{\gls@glossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUHead{#1}}{\memUHead{#1}}%
      \else
        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand{\gls@glossarymark}[1]{%
      \ifglsucmark
        \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
      \else
        \@mkboth{#1}{#1}%
      \fi
    }
  }
}
\providecommand{\glossarymark}[1]{%
  \ifglsucmark
    \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%

```

```

\else
  \@mkboth{#1}{#1}%
\fi
}
\newcommand*\setglossarysection[1]{%
\setkeys{glossaries.sty}{section=#1}}
\newcommand*\@glossarysection[2]{%
\ifdefempty\@glossarysecstar
  {%
    \csname\@glossarysec\endcsname[#1]{#2}%
  }%
  {%
    \csname\@glossarysec\endcsname*{#2}%
    \gls@toc{#1}{\@glossarysec}%
  }%
  \@glossaryseclabel
}
\newcommand*\@pglossarysection[2]{%
\gls@clearpage
\phantomsection
\ifdefempty\@glossarysecstar
  {%
    \csname\@glossarysec\endcsname{#2}%
  }%
  {%
    \gls@toc{#1}{\@glossarysec}%
    \csname\@glossarysec\endcsname*{#2}%
  }%
  \@glossaryseclabel
}
\newcommand*\gls@docclearpage{%
\ifthenelse{\equal{\@glossarysec}{chapter}}{%
  {%
    \ifcsundef{cleardoublepage}%
    {%
      \clearpage
    }%
    {%
      \ifcsdef{if@openright}%
      {%
        \if@openright
          \cleardoublepage
        \else
          \clearpage
        \fi
      }%
    }%
    \cleardoublepage
  }%
}
}

```



```

}%
{}%
}
\newcommand*\glsclearpage{\gls@doclearpage}
\newcommand*\@gls@toc}[2]{%
  \ifglstoc
  \ifglsnumberline
  \addcontentsline{toc}{#2}{\protect\numberline{#1}}%
  \else
  \addcontentsline{toc}{#2}{#1}%
  \fi
  \fi
}
\newcommand*\glsnoxindywarning}[1]{%
  \GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
}
\newcommand*\glsnomakeindexwarning}[1]{%
  \GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%
}
\ifglsxindy
  \edef\xdyattributes{\string"default\string"}%
\fi
\ifglsxindy
  \edef\xdyattributelist{}%
\fi
\ifglsxindy
  \def\xdylocref{}
\fi
\newcommand*\@gls@ifinlist}[4]{%
  \def\@do@ifinlist##1,#1,##2\end@doifinlist{%
    \def\@gls@listsuffix{##2}%
    \ifx\@gls@listsuffix\@empty
      #4%
    \else
      #3%
    \fi
  }%
  \@do@ifinlist,#2,#1,\end@doifinlist
}
\ifglsxindy
  \newcommand*\@xdycounters{\glscounter}
  \newcommand*\GlsAddXdyCounters[1]{%
    \@for\@gls@ctr:=#1\do{%
      \edef\@do@addcounter{%
        \noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
        {%
          \noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
            \noexpand\@gls@ctr}%
        }%
      }%
    }%
  }%

```

```

        \do@addcounter
    }
}
\@onlypremakeg\GlsAddXdyCounters
\else
\newcommand*\GlsAddXdyCounters[1]{%
\glsnoxywarning\GlsAddXdyAttribute
}
\fi
\newcommand*\@disabled@glssaddxdycounters{%
\PackageError{glossaries}{\string\GlsAddXdyCounters\space
can't be used after \string\GlsAddXdyAttribute}{Move all
occurrences of \string\GlsAddXdyCounters\space before the first
instance of \string\GlsAddXdyAttribute}%
}
\ifglxindy
\newcommand*\@glssaddxyattribute[2]{%
\edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string" ^^J
\string"#2#1\string"}%
\expandafter\toks@\expandafter{\@xdylocref}%
\edef\@xdylocref{\the\toks@ ^^J%
(markup-locref
:open \string"\glstildechar n%
\expandafter\string\csname glsX#2X#1\endcsname
\string" ^^J
:close \string"\string" ^^J
:attr \string"#2#1\string")}%
\expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
\setentrycounter[##1]{#2}\csname #1\endcsname{##2}%
}%
}
\newcommand*\GlsAddXdyAttribute[1]{%
\ifx\@xdyattributelist\@empty
\edef\@xdyattributelist{#1}%
\else
\edef\@xdyattributelist{\@xdyattributelist,#1}%
\fi
\@for\@this@counter:=\@xdycounters\do{%
\protected@edef\gls@do@addxyattribute{%
\noexpand\@glssaddxyattribute{#1}{\@this@counter}%
}
\gls@do@addxyattribute
}%
\let\GlsAddXdyCounters\@disabled@glssaddxdycounters
}
\@onlypremakeg\GlsAddXdyAttribute
\else
\newcommand*\GlsAddXdyAttribute[1]{%
\glsnoxywarning\GlsAddXdyAttribute}
\fi

```

```

\ifglxindy
\newcommand*{\@gls@addpredefinedattributes}{%
  \GlsAddXdyAttribute{glsnumberformat}
  \GlsAddXdyAttribute{textrm}
  \GlsAddXdyAttribute{textsf}
  \GlsAddXdyAttribute{texttt}
  \GlsAddXdyAttribute{textbf}
  \GlsAddXdyAttribute{textmd}
  \GlsAddXdyAttribute{textit}
  \GlsAddXdyAttribute{textup}
  \GlsAddXdyAttribute{textsl}
  \GlsAddXdyAttribute{textsc}
  \GlsAddXdyAttribute{emph}
  \GlsAddXdyAttribute{glsnumber}
  \GlsAddXdyAttribute{hyperm}
  \GlsAddXdyAttribute{hypersf}
  \GlsAddXdyAttribute{hypertt}
  \GlsAddXdyAttribute{hyperbf}
  \GlsAddXdyAttribute{hypermd}
  \GlsAddXdyAttribute{hyperit}
  \GlsAddXdyAttribute{hyperup}
  \GlsAddXdyAttribute{hypersl}
  \GlsAddXdyAttribute{hypersc}
  \GlsAddXdyAttribute{hyperemph}
  \GlsAddXdyAttribute{glsignore}
}
\else
  \let\@gls@addpredefinedattributes\relax
\fi
\def\@xdyuseralphabets{}
\ifglxindy
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \edef\@xdyuseralphabets{%
      \@xdyuseralphabets ^^J
      (define-alphabet "#1" (#2))}
  \else
    \newcommand*{\GlsAddXdyAlphabet}[2]{%
      \glsnoxindywarning\GlsAddXdyAlphabet}
  \fi
\ifglxindy
  \newcommand*{\@gls@xdy@locationlist}{%
    roman-page-numbers,%
    Roman-page-numbers,%
    arabic-page-numbers,%
    alpha-page-numbers,%
    Alpha-page-numbers,%
    Appendix-page-numbers,%
    arabic-section-numbers%
  }
\protected@edef\@gls@roman{\@roman{0}\string"

```

```

        \string"roman-numbers-lowercase\string" :sep \string}}}%
\@onelevel@sanitize\@gls@roman
\edef\@tmp{\string" \string"roman-numbers-lowercase\string"
:sep \string}}}%
\@onelevel@sanitize\@tmp
\ifx\@tmp\@gls@roman
  \expandafter
  \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
    \string"roman-numbers-lowercase\string"%
  }%
\else
  \expandafter
  \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{
    :sep \string"\@gls@roman\string"%
  }%
\fi
\expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
  \string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
  \string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
  \string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
  \string"ALPHA\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
  \string"ALPHA\string"
  :sep \string"\@gls@Alphacompositor\string"
  \string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
  \string"arabic-numbers\string"
  :sep \string"\glscompositor\string"
  \string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglxindy
\newcommand\@xdycrossrefhook{}
\fi
\ifglxindy
\newcommand*\GlsAddXdyLocation}[3][[]]{%
  \def\@gls@tmp{#1}%
  \ifx\@gls@tmp\@empty
    \edef\@xdyuserlocationdefs{%
      \@xdyuserlocationdefs ^^J%
    }
  \fi

```

```

        (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]{%
\glsnoinxindwarning\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]{%
\glsnoinxindwarning\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
  \ifglxsxindy@glnumbers
    \def\@xdylettergroups{(define-letter-group
      \string"glnumbers\string"^^J\space\space\space
      :prefixes (\string"0\string" \string"1\string"
        \string"2\string" \string"3\string" \string"4\string"
        \string"5\string" \string"6\string" \string"7\string"
        \string"8\string" \string"9\string")^^J\space\space\space
      \@xdynumbergrouporder)}
    \else
      \def\@xdylettergroups{}
    \fi
  \fi
\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]{%
  \edef\@glo@list{\csname glolist@#1\endcsname}%
  \@for#2:=\@glo@list\do
  {%
    \ifdefempty{#2}{#3}%
  }%
}
\newcommand*\forallglsentries}[3][\@glo@types]{%
  \expandafter\forallglossaries\expandafter[#1]{\@@this@glo@}%
  {%
    \forallglsentries[\@@this@glo@]{#2}{#3}%
  }%
}
\newcommand{\ifglossaryexists}{%
  \@ifstar\s@ifglossaryexists\@ifglossaryexists
}
\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{@glo@type@#1@out}{#3}{#2}%
}
\newcommand{\s@ifglossaryexists}[3]{%
  \ifcsundef{glolist@#1}{#3}{#2}%
}
\newcommand*\glsdetoklabel}[1]{#1}
\newcommand{\ifglsentryexists}[3]{%

```

```

\ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}
\newcommand*{\ifglsused}[3]{%
\ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}
\newcommand{\glsdoifexists}[2]{%
\ifglsentryexists{#1}{#2}{%
\PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
has not been defined}{You need to define a glossary entry before you
can use it.}}%
}
\newcommand{\glsdoifnoexists}[2]{%
\ifglsentryexists{#1}{%
\PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
been defined}{}}#2}%
}
\newcommand{\glsdoifexistsorwarn}[2]{%
\ifglsentryexists{#1}{#2}{%
\GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
has not been defined}%
}%
}
\newcommand{\glsdoifexistsordo}[3]{%
\ifglsentryexists{#1}{#2}{%
\PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
has not been defined}{You need to define a glossary entry before you
can use it.}}%
#3%
}%
}
\newcommand{\doifglossarynoexistsordo}[3]{%
\ifglossaryexists*{#1}%
{%
\PackageError{glossaries}{Glossary type ‘#1’ already exists}{}%
#3%
}%
#2}%
}
\newrobustcmd{\ifglshaschildren}[3]{%
\glsdoifexists{#1}%
{%
\def\do@glshaschildren{#3}%
\edef\@gls@thislabel{\glsdetoklabel{#1}}%
\expandafter\forglsentries\expandafter
[\csname glo@\@gls@thislabel @type\endcsname]
{\glo@label}%
{%
\letcs\glo@parent{glo@\glo@label @parent}%
\ifdefequal\@gls@thislabel\glo@parent
{%

```



```

        \def\do@glshaschildren{#2}%
        \@endfortrue
    }%
    {}%
    }%
    \do@glshaschildren
}
}
\newcommand{\ifglshasparent}[3]{%
    \glstoifexists{#1}%
    {%
        \ifcsempy{glo@\glstetoklabel{#1}@parent}{#3}{#2}%
    }%
}
\newcommand*{\ifglshasdesc}[3]{%
    \ifcsempy{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}@#1}%
    \ifdefempty\@glo@thisvalue
    {%
      #4%
    }%
  }%
  {%
    \ifdefequal\@glo@thisvalue\@gls@default@value
    {%
      #4%
    }%
  }%
  \let\glscurrentfieldvalue\@glo@thisvalue
  #3%
}%
}%
}%
\GlossariesWarning{Unknown entry field '#1'}%
#4%
}%
}%

```

```

}%
}
\newcommand*\glscurrentfieldvalue{}
\newcommand*\@glo@types{,}
\newcommand*\@gls@provide@newglossary{%
  \protected@write\@auxout{}{\string\providecommand\string\@newglossary[4]{}}%
  \let\@gls@provide@newglossary\relax
}
\newcommand*\defglsentryfmt}[2][\glsdefaulttype]{%
  \csgdef{gls@#1@entryfmt}{#2}%
}
\newcommand*\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*\@gls@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\glsdefaulttype
    {%
      \gdef\glsdefaulttype{#2}%
    }-}%
  \toks@{#2}\edef\@glo@types{\@glo@types\the\toks@,}%
  \expandafter\gdef\csname glolist@#2\endcsname{,}%
  \expandafter\edef\csname @glotype@#2@log\endcsname{#1}%
  \expandafter\edef\csname @glotype@#2@in\endcsname{#3}%
  \expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
  \expandafter\@gls@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}%
  \ifglsxindy
    \GlsAddXdyCounters{#2}%
  \fi
}
\newcommand*\@gls@getcounter}[1]{%
  \csname @glotype@#1@counter\endcsname
}
\glsdefmain
\@gls@do@acronymsdef
\@gls@do@symbolsdef
\@gls@do@numbersdef
\@gls@do@indexdef
\newcommand*\newignoredglossary}[1]{%
  \ifdefempty\@ignored@glossaries
  {%
    \edef\@ignored@glossaries{#1}%
  }%
  {%
    \eappto\@ignored@glossaries{,#1}%
  }%
  \csgdef{glolist@#1}{,}%
  \ifcsundef{gls@#1@entryfmt}%
  {%
    \defglsentryfmt[#1]{\glsentryfmt}%
  }%
  {}%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*\@gls@nohyperlist}{#1}%
  }%

```

```

    {%
      \eappto\@gls@nohyperlist{,#1}%
    }%
  }
  \newcommand*\@ignored@glossaries{}
  \newcommand*\ifignoredglossary}[3]{%
    \edef\@gls@igtype{#1}%
    \expandafter\DTLifinlist\expandafter
      {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
  }
  \define@key{glossentry}{name}{%
    \def\@glo@name{#1}%
  }
  \define@key{glossentry}{description}{%
    \def\@glo@desc{#1}%
  }
  \define@key{glossentry}{descriptionplural}{%
    \def\@glo@descplural{#1}%
  }
  \define@key{glossentry}{sort}{%
    \def\@glo@sort{#1}}
  \define@key{glossentry}{text}{%
    \def\@glo@text{#1}%
  }
  \define@key{glossentry}{plural}{%
    \def\@glo@plural{#1}%
  }
  \define@key{glossentry}{first}{%
    \def\@glo@first{#1}%
  }
  \define@key{glossentry}{firstplural}{%
    \def\@glo@firstplural{#1}%
  }
  \newcommand*\@gls@default@value{\relax}
  \define@key{glossentry}{symbol}{%
    \def\@glo@symbol{#1}%
  }
  \define@key{glossentry}{symbolplural}{%
    \def\@glo@symbolplural{#1}%
  }
  \define@key{glossentry}{type}{%
    \def\@glo@type{#1}}
  \define@key{glossentry}{counter}{%
    \ifcsundef{c@#1}%
      {%
        \PackageError{glossaries}%
          {There is no counter called ‘#1’}%
          {%
            The counter key should have the name of a valid counter
            as its value%
          }
      }
    }

```

```

    }%
  }%
  {%
    \def\@glo@counter{#1}%
  }%
}
\define@key{glossentry}{see}{%
  \gls@set@xr@key{see}{\@glo@see}{#1}%
}
\newcommand*\@gls@set@xr@key[3]{%
  \renewcommand*\@gls@xr@key{#1}%
  \gls@checkseeallowed
  \def#2{#3}%
  \@glo@seeautonumberlist
}
\newcommand*\@gls@xr@key{see}
\newcommand*\@gls@checkseeallowed{%
  \@gls@see@noindex
}
\newcommand*\@gls@checkseeallowed@preambleonly{%
  \GlossariesWarning{glossaries}%
  {'\@gls@xr@key' key doesn't have any effect when used in the document
  environment. Move the definition to the preamble
  after \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@key{glossentry}{parent}{%
  \def\@glo@parent{#1}}
\define@choicekey{glossentry}{nonumberlist}%
  [\@gls@nonumberlist@val\@gls@nonumberlist@nr]{true,false}[true]%
{%
  \ifcase\@gls@nonumberlist@nr\relax
    \def\@glo@prefix{\@gls@nonextpages}%
    \@gls@savenonumberlist{true}%
  \else
    \def\@glo@prefix{\@gls@nextpages}%
    \@gls@savenonumberlist{false}%
  \fi
}
\newcommand*\@gls@savenonumberlist[1]{}
\newcommand*\@gls@initnonumberlist{}%
\newcommand*\@gls@storenonumberlist[1]{}
\newcommand*\@gls@enablesavenonumberlist{%
  \renewcommand*\@gls@initnonumberlist{}%
  \undef\@glo@nonumberlist
}%
\renewcommand*\@gls@savenonumberlist[1]{%
  \def\@glo@nonumberlist{##1}%
}%
\renewcommand*\@gls@storenonumberlist[1]{%

```

```

\ifdef\@glo@nonumberlist
{%
  \cslet{glo\@glsdetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
}%
{}%
}%
\appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}
\define@key{glossentry}{user1}{%
  \def\@glo@useri{#1}%
}
\define@key{glossentry}{user2}{%
  \def\@glo@userii{#1}%
}
\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#1}%
}
\define@key{glossentry}{short}{%
  \def\@glo@short{#1}%
}
\define@key{glossentry}{shortplural}{%
  \def\@glo@shortpl{#1}%
}
\define@key{glossentry}{long}{%
  \def\@glo@long{#1}%
}
\define@key{glossentry}{longplural}{%
  \def\@glo@longpl{#1}%
}
\newcommand*\@glsnoname}{%
  \PackageError{glossaries}{name key required in
  \string\newglossaryentry\space for entry '\@glo@label'}{You
  haven't specified the entry name}}
\newcommand*\@glsnodesc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '\@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}

```

```

}%
}%
\newcommand*{\@glsdefaultplural}{}
\newcommand*{\@gls@missingnumberlist}[1]{%
??%
\ifgls@savenumberlist
\GlossariesWarning{Missing number list for entry '#1'.
Maybe makeglossaries + rerun required}%
\else
\PackageError{glossaries}%
{Package option 'savenumberlist=true' required}%
{%
You must use the 'savenumberlist' package option
to reference location lists.%
}%
\fi
}
\newcommand*{\@glsdefaultsort}{\@glo@name}
\newcount\gls@level
\newcommand{\@gls@noexpand@field}[3]{%
\expandafter\global\expandafter
\let\csname glo@#1@#2\endcsname#3%
}
\newcommand{\@gls@noexpand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\csuse{gls@assign@#3@field}{#2}{#4}%
}%
}%
}%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\@gls@noexpand@field{#2}{#3}{\@gls@value}%
}%
{%
\@gls@noexpand@field{#2}{#3}{#4}%
}%
}%
}
\newcommand{\@gls@expand@field}[3]{%
\expandafter
\protected@xdef\csname glo@#1@#2\endcsname{#3}%
}

```



```

\newcommand{\@gls@expand@fields}[4]{%
  \ifcsdef{gls@assign@#3@field}
  {%
    \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\gls@checkseeallowed\gls@checkseeallowed@preambleonly
  \let\newglossaryentry\new@glossaryentry
}

```

```

\newrobustcmd{\provideglossaryentry}[2]{%
  \ifglentryexists{#1}%
  {}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\@onlypreamble{\provideglossaryentry}
\newrobustcmd{\new@glossaryentry}[2]{%
  \ifundef\@gls@deffile
  {%
    \global\newwrite\@gls@deffile
    \immediate\openout\@gls@deffile=\jobname.glsdefs
  }%
  {}%
  \ifglentryexists{#1}{}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
  \@gls@writedef{#1}%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*\@gls@begindocdefs{%
  \@gls@enablesavenonumberlist
  \edef\@gls@restoreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
  \makeatletter
  \InputIfFileExists{\jobname.glsdefs}{%}{%
  \@gls@restoreat
  \undef\@gls@restoreat
  \gls@defdocnewglossaryentry
}
\newcommand*\@gls@writedef}[1]{%
  \immediate\write\@gls@deffile
  {%
    \string\ifglentryexists{#1}{}\glspercentchar^^J%
    \expandafter\@gobble\string\{\glspercentchar^^J%
    \string\gls@defglossaryentry{\glsdetoklabel{#1}}\glspercentchar^^J%
    \expandafter\@gobble\string\{\glspercentchar%
  }%
  \@for\@gls@map:=\@gls@keymap\do
  {%
    \letcs\glo@value{glo@\glsdetoklabel{#1}}\expandafter\@secondoftwo\@gls@map}%
    \ifdef\glo@value
    {%
      \@onelevel@sanitize\glo@value
      \immediate\write\@gls@deffile
      {%
        \expandafter\@firstoftwo\@gls@map
        =\expandafter\@gobble\string\{\glo@value\expandafter\@gobble\string\},%
      }%
    }%
  }%
}

```

```

        \glspercentchar
    }%
  }%
  {}%
}%
\gls.writedefhook
\immediate\write\@gls@deffile
{%
    \glspercentchar^^J%
    \expandafter\@gobble\string\}\glspercentchar^^J%
    \expandafter\@gobble\string\}\glspercentchar%
}%
}
\newcommand*\@gls@keymap{%
  {name}{name},%
  {sort}{sortvalue},% unescaped sort value
  {type}{type},%
  {first}{first},%
  {firstplural}{firstpl},%
  {text}{text},%
  {plural}{plural},%
  {description}{desc},%
  {descriptionplural}{descplural},%
  {symbol}{symbol},%
  {symbolplural}{symbolplural},%
  {user1}{useri},%
  {user2}{userii},%
  {user3}{useriii},%
  {user4}{useriv},%
  {user5}{userv},%
  {user6}{uservi},%
  {long}{long},%
  {longplural}{longpl},%
  {short}{short},%
  {shortplural}{shortpl},%
  {counter}{counter},%
  {parent}{parent}%
}
\newcommand*\@gls@fetchfield}[2]{%
  \edef\@gls@thisval{#2}%
  \for\@gls@map:=\@gls@keymap\do{%
    \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
    \ifdefequal{\@this@key}{\@gls@thisval}%
    {%
      \edef#1{\expandafter\@secondoftwo\@gls@map}%
      \@endfortrue
    }%
  }%
}%
}

```

```

\newcommand*\glsaddstoragekey}{\ifstar\sglsaddstoragekey\glsaddstoragekey}
\newcommand*\@sglsaddstoragekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
  {}%
  \@glsaddstoragekey{#1}%
}
\newcommand*\@glsaddstoragekey}[3]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{, {#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
    \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
  }%
  {}%
  \PackageError{glossaries}{Key ‘#1’ already exists}{}%
  }%
}
\newcommand*\glsaddkey}{\ifstar\sglsaddkey\glsaddkey}
\newcommand*\@sglsaddkey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
  {}%
  \@glsaddkey{#1}%
}
\newcommand*\@glsaddkey}[7]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{, {#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
  }%
}

```

```

\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
\newcommand*{#4}[1]{\@Gls@entry@field{##1}{#1}}%
\ifcsdef{@gls@user@#1@}%
{
  \PackageError{glossaries}%
  {Can't define '\string#5' as helper command
   '\expandafter\string\csname @gls@user@#1@\endcsname' already exists}%
  {}%
}%
{
  \expandafter\newcommand\expandafter*\expandafter
  {\csname @gls@user@#1@\endcsname}[2][]{%
   \new@ifnextchar[%
    {\csuse{@gls@user@#1@}{##1}{##2}}%
    {\csuse{@gls@user@#1@}{##1}{##2}[[]]}%
  \csdef{@gls@user@#1@}##1##2[##3]{%
   \@gls@field@link{##1}{##2}{#3{##2}##3}%
  }%
  \newrobustcmd*{#5}{%
   \expandafter\@gls@hyp@opt\csname @gls@user@#1@\endcsname}%
  }%
\ifcsdef{@Gls@user@#1@}%
{
  \PackageError{glossaries}%
  {Can't define '\string#6' as helper command
   '\expandafter\string\csname @Gls@user@#1@\endcsname' already exists}%
  {}%
}%
{
  \expandafter\newcommand\expandafter*\expandafter
  {\csname @Gls@user@#1@\endcsname}[2][]{%
   \new@ifnextchar[%
    {\csuse{@Gls@user@#1@}{##1}{##2}}%
    {\csuse{@Gls@user@#1@}{##1}{##2}[[]]}%
  \csdef{@Gls@user@#1@}##1##2[##3]{%
   \@gls@field@link{##1}{##2}{#4{##2}##3}%
  }%
  \newrobustcmd*{#6}{%
   \expandafter\@gls@hyp@opt\csname @Gls@user@#1@\endcsname}%
  }%
\ifcsdef{@GLS@user@#1@}%
{
  \PackageError{glossaries}%
  {Can't define '\string#7' as helper command
   '\expandafter\string\csname @GLS@user@#1@\endcsname' already exists}%
  {}%
}%
{
  \expandafter\newcommand\expandafter*\expandafter
  {\csname @GLS@user@#1@\endcsname}[2][]{%

```

```

        \new@ifnextchar [%
            {\csuse{@GLS@user@#1@}{##1}{##2}}%
            {\csuse{@GLS@user@#1@}{##1}{##2}[]}%
        \csdef{@GLS@user@#1@}##1##2[##3]{%
            \@gls@field@link{##1}{##2}{\mfirstucMakeUppercase{#3{##2}##3}}%
        }%
        \newrobustcmd*{#7}{%
            \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
        }%
    }%
}%
{
    \PackageError{glossaries}{Key ‘#1’ already exists}{%
}
}
\newcommand{\glsfieldxdef}[3]{%
\glsdoifexists{#1}%
{%
    \edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
        \protected@csxdef{glo@\@glo@label @#2}{#3}%
    }%
    {%
        \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{%
    }%
}
}
\newcommand{\glsfielddedef}[3]{%
\glsdoifexists{#1}%
{%
    \edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
        \protected@csedef{glo@\@glo@label @#2}{#3}%
    }%
    {%
        \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{%
    }%
}
}
}
\newcommand{\glsfielddgdef}[3]{%
\glsdoifexists{#1}%
{%
    \edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
        \expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
    }%
    {%
        \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{%
}
}
}

```

```

    }%
  }%
}
\newcommand{\glsfielddef}[3]{%
\glsdoifexists{#1}%
{%
  \edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand{\glsfieldfetch}[3]{%
\glsdoifexists{#1}%
{%
  \edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \letcs#3{glo@\@glo@label @#2}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand{\ifglsfieldeq}[5]{%
\glsdoifexists{#1}%
{%
  \edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand{\ifglsfielddefeq}[5]{%
\glsdoifexists{#1}%
{%
  \edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \expandafter\ifdefstrequal
      \csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
  }%
}

```

```

}%
{%
  \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
}%
}%
}
\newcommand{\ifglsfieldcseq}[5]{%
\glsdoifexists{#1}%
{%
  \edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \ifcsstrequal{glo@\@glo@label @#2}{#3}{#4}{#5}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand*{\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\@glo@list@\csname glo@list@\@glo@type\endcsname}%
\expandafter\xdef\csname glo@list@\@glo@type\endcsname{%
    \@glo@list@\@glo@label},}%
}%
\gls@level=0\relax
\ifx\@glo@parent\@empty
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{%
\else
    \ifdefequal\@glo@label\@glo@parent%
        {%
            \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
            \def\@glo@parent{%
            \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{%
            }%
            {%
                \ifglsentryexists{\@glo@parent}%
                {%
                    \expandafter\xdef\csname glo@\@glo@label @parent\endcsname{%
                        \@glo@parent}%
                    \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
                    \advance\gls@level by 1\relax
                    \ifx\@glo@name\@glsnoname
                        \expandafter\let\expandafter\@glo@name
                        \csname glo@\@glo@parent @name\endcsname
                        \ifx\@glo@plural\@gls@default@value
                            \expandafter\let\expandafter\@glo@plural
                            \csname glo@\@glo@parent @plural\endcsname
                        \fi
                    \fi
                }%
            }%
            \PackageError{glossaries}%
            {%
                Invalid parent ‘\@glo@parent’
                for entry ‘\@glo@label’ - parent doesn’t exist%
            }%
            {%
                Parent entries must be defined before their children%
            }%
            \def\@glo@parent{%
            \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{%
            }%
            }%
        }%
    \fi
\expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%

```

```

\gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
\letcs@glo@sort{glo@\@glo@label @sortvalue}%
\gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @text\endcsname\glspluralsuffix}%
  {\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @text\endcsname}%
  {\@glo@label}{first}{\@glo@first}%
\ifx@glo@first@gls@default@value
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @plural\endcsname}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\else
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\fi
\ifcsundef{glo@type@\@glo@type @counter}%
{%
  \def@glo@defaultcounter{\glscounter}%
}%
{%
  \letcs@glo@defaultcounter{glo@type@\@glo@type @counter}%
}%
\gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
\gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
\gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
\gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
\gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
\gls@assign@field{}{\@glo@label}{userv}{\@glo@userv}%
\gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
\gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
\gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
\gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
\gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%
\ifx@glo@name@glsnoname
  \glsnoname
  \let@glo@name@gls@default@value
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%
\ifcsundef{glo@\@glo@label @numberlist}%
{%
  \csxdef{glo@\@glo@label @numberlist}{%
    \noexpand@gls@missingnumberlist{\@glo@label}}%
}%
{%
  \gls@storenonumberlist{\@glo@label}%
  \def@glo@desc{\@glo@first}%
  \ifx@glo@desc@glo@desc

```

```

\let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@glsnodesc
\@glsnodesc
\let\@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
\@newglossaryentry@posthook
}
\newcommand*{\@glo@autosee}{%
\ifdefvoid\@glo@see{%
{%
\protected@edef\@do@glssee{%
\noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
\noexpand\expandafter\noexpand\@glssee\noexpand\@glo@list{\@glo@label}}%
\do@glssee
}%
\@glo@autoseehook
}%
\newcommand*{\@glo@autoseehook}{%

```

```

\newcommand*{\@newglossaryentryprehook}{}
\newcommand*{\@newglossaryentryposthook}{}
\newcommand*{\@newglossaryentry@defcounters}{}
\newcommand*{\glsmoveentry}[2]{%
  \edef\@glo@thislabel{\glsdetoklabel{#1}}%
  \edef\@glo@type{\csname glo@\@glo@thislabel @type\endcsname}%
  \def\@glo@list{,%}
  \for@gl@entries[\@glo@type]{\@glo@label}%
  {%
    \if@defequal\@glo@thislabel\@glo@label
      {}{\@eappto\@glo@list{\@glo@label,}}%
    }%
  \cslet\@glo@list[\@glo@type]{\@glo@list}%
  \csdef{glo@\@glo@thislabel @type}{#2}%
}
\if@gl@xindy
  \newcommand*{\@glossaryentryfield}{\string\@glossentry}
\else
  \newcommand*{\@glossaryentryfield}{\string@glossentry}
\fi
\if@gl@xindy
  \newcommand*{\@glossarysubentryfield}{%
    \string\@subglossentry}
\else
  \newcommand*{\@glossarysubentryfield}{%
    \string@subglossentry}
\fi
\newcommand{\@glo@storeentry}[1]{%
  \edef\@glo@esclabel{#1}%
  \@gls@checkmkidxchars\@glo@esclabel
  \protected@edef\@glo@sort{\csname glo@#1@sort\endcsname}%
  \@gls@checkmkidxchars\@glo@sort
  \@gls@checkmkidxchars\@glo@prefix
  \edef\@glo@parent{\csname glo@#1@parent\endcsname}%
  \if@gl@xindy
    \ifx\@glo@parent\@empty
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        (\string"\@glo@sort\string" %
        \string"\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %
      }%
    \else
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \csname glo@\@glo@parent @index\endcsname
        (\string"\@glo@sort\string" %
        \string"\@glo@prefix\@glossarysubentryfield
        {\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
      }%
    \fi
  \else
    \ifx\@glo@parent\@empty

```

```

        \onelevel@sanitize\@glo@prefix
        \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
            \@glo@sort\@gls@actualchar\@glo@prefix
            \@glossaryentryfield{\@glo@esclabel}%
        }%
    \else
        \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
            \csname glo@\@glo@parent @index\endcsname\@gls@levelchar
            \@glo@sort\@gls@actualchar\@glo@prefix
            \@glossarysubentryfield
                {\csname glo@#1@level\endcsname}{\@glo@esclabel}%
            }%
        \fi
    \fi
}
\AtBeginDocument{%
    \ifpackageloaded{amsmath}%
    {\let\gls@ifnotmeasuring\@gls@ifnotmeasuring}%
    {}%
}
\newcommand*\@gls@ifnotmeasuring[1]{%
    \ifmeasuring@
    \else
    #1%
    \fi
}
\newcommand*\gls@ifnotmeasuring[1]{#1}
\def\@gls@patchtabularx#1\hbox#2#3!!{%
    \def\TX@trial##1{#1\hbox{\let\glsunset\@gobble#2}#3}%
}
\newcommand*\glspatchtabularx{%
    \ifdef\TX@trial
    {%
        \expandafter\@gls@patchtabularx\TX@trial{##1}!!%
        \let\glspatchtabularx\relax
    }%
    {}%
}
\newcommand*\glsreset[1]{%
    \gls@ifnotmeasuring
    {%
        \glsdoifexists{#1}%
        {%
            \@glsreset{#1}%
        }%
    }%
}
\newcommand*\glslocalreset[1]{%
    \gls@ifnotmeasuring
    {%

```

```

\glsdoifexists{#1}%
{%
  \@glslocalreset{#1}%
}%
}%
}
\newcommand*\glsunset}[1]{%
\gls@ifnotmeasuring
{%
  \glsdoifexists{#1}%
  {%
    \@glsunset{#1}%
  }%
}%
}
\newcommand*\glslocalunset}[1]{%
\gls@ifnotmeasuring
{%
  \glsdoifexists{#1}%
  {%
    \@glslocalunset{#1}%
  }%
}%
}
\newcommand*\@glslocalunset{\@glslocalunset}
\newcommand*\@glslocalunset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
}
\newcommand*\@glsunset{\@glsunset}
\newcommand*\@glsunset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
}
\newcommand*\@glslocalreset{\@glslocalreset}
\newcommand*\@glslocalreset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
}
\newcommand*\@glsreset{\@glsreset}
\newcommand*\@glsreset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
}
\newcommand*\glsresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsreset{\@glsentry}%
  }%
}
\newcommand*\glslocalresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalreset{\@glsentry}%
  }%
}

```

```

}%
}
\newcommand*\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsunset{\@glsentry}%
  }%
}
\newcommand*\glslocalunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalunset{\@glsentry}%
  }%
}
\newcommand*\@@newglossaryentry@defcounters{%
  \csdef{glo@\@glo@label @currcount}{0}%
  \csdef{glo@\@glo@label @prevcount}{0}%
}
\newcommand*\glsenableentrycount{%
  \let\@newglossaryentry@defcounters\@@newglossaryentry@defcounters
  \renewcommand*\gls@defdocnewglossaryentry{%
    \renewcommand*newglossaryentry[2]{%
      \PackageError{glossaries}{\string\newglossaryentry\space
        may only be used in the preamble when entry counting has
        been activated}{If you use \string\glsenableentrycount\space
        you must place all entry definitions in the preamble not in
        the document environment}%
    }%
  }%
}
\newcommand*\glsentrycurrcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
  {0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*\glsentryprevcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
  {0}{\@gls@entry@field{##1}{prevcount}}%
}%
\renewcommand*\@glsunset}[1]{%
  \@glsunset{##1}%
  \@gls@increment@currcount{##1}%
}%
\renewcommand*\@glslocalunset}[1]{%
  \@glslocalunset{##1}%
  \@gls@local@increment@currcount{##1}%
}%
\renewcommand*\@glsreset}[1]{%
  \@glsreset{##1}%
  \csgdef{glo@\glsdetoklabel{##1}@currcount}{0}%
}%
\renewcommand*\@glslocalreset}[1]{%

```



```

    \@glslocalreset{##1}%
    \csdef{glo@glsdetoklabel{##1}@currcount}{0}%
  }%
\def\@cgl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cglformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@gls@{##1}{##2}[##3]%
  \fi
}%
\def\@cgl@spl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cgl@splformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@gls@{##1}{##2}[##3]%
  \fi
}%
\def\@cGls@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cGlsformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@Gls@{##1}{##2}[##3]%
  \fi
}%
\def\@cGls@spl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cGls@splformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@Gls@{##1}{##2}[##3]%
  \fi
}%
\AtEndDocument{\@gls@write@entrycounts}%
\renewcommand*{\@gls@entry@count}[2]{%
  \csgdef{glo@glsdetoklabel{##1}@prevcount}{##2}%
}%
\let\glsenableentrycount\relax
}
\@onlypreamble\glsenableentrycount
\newcommand*{\@gls@increment@currcount}[1]{%
  \csxdef{glo@glsdetoklabel{##1}@currcount}{%
    \number\numexpr\glsentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@local@increment@currcount}[1]{%
  \csedef{glo@glsdetoklabel{##1}@currcount}{%
    \number\numexpr\glsentrycurrcount{##1}+1}%
}
}

```

```

\newcommand*\@gls@write@entrycounts}{%
  \immediate\write\@auxout
  {\string\providecommand*\string\@gls@entry@count}[2]{}}%
\forallglsentries{\@glsentry}{%
  \ifglsused{\@glsentry}%
  {\immediate\write\@auxout
    {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}}%
  }%
}%
}
\newcommand*\@gls@entry@count}[2]{%
\newrobustcmd*\@cgl{s}{\@gls@hyp@opt\@cgl{s}}
\newcommand*\@cgl{s}[2][{}]{%
  \new@ifnextchar[{\@cgl{s}@{#1}{#2}}]{\@cgl{s}@{#1}{#2}}[{}]}%
}
\def\@cgl{s}@#1#2[#3]{%
  \GlossariesWarning{\string\cgl{s}\space is defaulting to
    \string\gls\space since you haven't enabled entry counting}%
  \@cgl{s}@{#1}{#2}[#3]}%
}
\newcommand*\@cgl{sformat}[2]{%
  \ifglsashaslong{#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]{%
  \ifglsashaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}#2%
}
\newrobustcmd*\@cgl{spl}{\@gls@hyp@opt\@cgl{spl}}
\newcommand*\@cgl{spl}[2][{}]{%
  \new@ifnextchar[{\@cgl{spl}@{#1}{#2}}]{\@cgl{spl}@{#1}{#2}}[{}]}%
}
\def\@cgl{spl}@#1#2[#3]{%
  \GlossariesWarning{\string\cgl{spl}\space is defaulting to
    \string\glspl\space since you haven't enabled entry counting}%
  \@cgl{spl}@{#1}{#2}[#3]}%
}
\newcommand*\@cgl{splformat}[2]{%
  \ifglsashaslong{#1}{\glsentrylongpl{#1}}{\glsentryfirstplural{#1}}#2%
}
\newrobustcmd*\@cGlspl{\@gls@hyp@opt\@cGlspl}
\newcommand*\@cGlspl}[2][{}]{%
  \new@ifnextchar[{\@cGlspl@{#1}{#2}}]{\@cGlspl@{#1}{#2}}[{}]}%
}

```

```

}
\def\cGlspl@#1#2[#3]{%
  \GlossariesWarning{\string\cGlspl\space is defaulting to
    \string\Glspl\space since you haven't enabled entry counting}%
  \@Glspl@{#1}{#2}[#3]%
}
\newcommand*\cGlsplformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
}
\newcommand*\loadglsentries}[2][\@gls@default]{%
  \let\@gls@default\glsdefaulttype
  \def\glsdefaulttype{#1}\input{#2}%
  \let\glsdefaulttype\@gls@default
}
\@onlypreamble{\loadglsentries}
\newcommand*\glsformat}[1]{#1}
\newcommand*\glsentryfmt}{%
  \@@gls@default@entryfmt\glsdisplayfirst\glsdisplay
}
\newcommand*\@@gls@default@entryfmt}[2]{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          #2{\glsentryplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
          }%
        {%
          #1{\glsentryfirstplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
          }%
        }%
      }%
    }%
    \ifglsused\glslabel
    {%
      \ifbool{glscompatible-3.07}%
      {%
        \protected@edef\@glo@etext{%
          #2{\glsentryplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
          \xmakefirstuc\@glo@etext
        }%
      }%
    }%
  }%
}

```

```

#2{\Glsentryplural{\glslabel}}%
{\glsentrydescplural{\glslabel}}%
{\glsentrysymbolplural{\glslabel}}{\glsinsert}%
}%
}%
{%
\ifbool{glscompatible-3.07}%
{%
\protected@edef\@glo@etext{%
#1{\Glsentryfirstplural{\glslabel}}%
{\glsentrydescplural{\glslabel}}%
{\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
\xmakefirstuc\@glo@etext
}%
{%
#1{\Glsentryfirstplural{\glslabel}}%
{\glsentrydescplural{\glslabel}}%
{\glsentrysymbolplural{\glslabel}}{\glsinsert}%
}%
}%
}%
}%
\ifglsused\glslabel
{%
\mfirstucMakeUppercase{#2{\glsentryplural{\glslabel}}%
{\glsentrydescplural{\glslabel}}%
{\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
}%
{%
\mfirstucMakeUppercase{#1{\glsentryfirstplural{\glslabel}}%
{\glsentrydescplural{\glslabel}}%
{\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
}%
}%
}%
\glscapscase
{%
\ifglsused\glslabel
{%
#2{\glsentrytext{\glslabel}}%
{\glsentrydesc{\glslabel}}%
{\glsentrysymbol{\glslabel}}{\glsinsert}%
}%
{%
#1{\glsentryfirst{\glslabel}}%
{\glsentrydesc{\glslabel}}%
{\glsentrysymbol{\glslabel}}{\glsinsert}%
}%
}%

```

```

{%
  \ifglsused\glslabel
  {%
    \ifbool{glscompatible-3.07}%
    {%
      \protected@edef\@glo@etext{%
        #2{\glsentrytext{\glslabel}}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{\glsinsert}}%
      \xmakefirstuc\@glo@etext
    }%
    {%
      #2{\Glsentrytext{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
  }%
  {%
    \ifbool{glscompatible-3.07}%
    {%
      \protected@edef\@glo@etext{%
        #1{\glsentryfirst{\glslabel}}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{\glsinsert}}%
      \xmakefirstuc\@glo@etext
    }%
    {%
      #1{\Glsentryfirst{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
  }%
}
}
}
{%
  \ifglsused\glslabel
  {%
    \mfirstucMakeUppercase{#2{\glsentrytext{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}}%
  }%
  {%
    \mfirstucMakeUppercase{#1{\glsentryfirst{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}}%
  }%
}
}
}
{%
  \ifglsused{\glslabel}%

```

```

    {%
      #2{\glscustomtext}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{}}%
    }%
    {%
      #1{\glscustomtext}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{}}%
    }%
  }%
}
\newcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          \glsentryplural{\glslabel}\glsinsert
        }%
        {%
          \glsentryfirstplural{\glslabel}\glsinsert
        }%
      }%
    }%
    {%
      \ifglsused\glslabel
      {%
        \Glsentryplural{\glslabel}\glsinsert
      }%
      {%
        \Glsentryfirstplural{\glslabel}\glsinsert
      }%
    }%
    {%
      \ifglsused\glslabel
      {%
        \mfirstucMakeUppercase
        {\glsentryplural{\glslabel}\glsinsert}%
      }%
      {%
        \mfirstucMakeUppercase
        {\glsentryfirstplural{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glscapscase
  }%
}

```

```

    {%
      \ifglused\glslabel
      {%
        \glsentrytext{\glslabel}\glsinsert
      }%
      {%
        \glsentryfirst{\glslabel}\glsinsert
      }%
    }%
    {%
      \ifglused\glslabel
      {%
        \Glsentrytext{\glslabel}\glsinsert
      }%
      {%
        \Glsentryfirst{\glslabel}\glsinsert
      }%
    }%
    {%
      \ifglused\glslabel
      {%
        \mfirstucMakeUppercase{\glsentrytext{\glslabel}\glsinsert}%
      }%
      {%
        \mfirstucMakeUppercase{\glsentryfirst{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glscustomtext\glsinsert
  }%
}
\newcommand*{\glsgenacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \acronymfont{\glsentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \mfirstucMakeUppercase
            {\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert}%
        }%
      }%
    }%
  }%
}

```



```

\newcommand*\genacrfullformat}[2]{%
  \glsentrylong{#1}#2\space
  (\protect\firstacronymfont{\glsentryshort{#1}})%
}
\newcommand*\Genacrfullformat}[2]{%
  \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
  \xmakefirstuc\gls@text
}
\newcommand*\genplacrfullformat}[2]{%
  \glsentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glsentryshortpl{#1}})%
}
\newcommand*\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \xmakefirstuc\gls@text
}
\newcommand*\glsdisplayfirst}[4]{#1#4}
\newcommand*\glsdisplay}[4]{#1#4}
\newcommand*\defglsdisplay}[2][\glsdefaulttype]{%
  \GlossariesWarning{\string\defglsdisplay\space is now obsolete.^^J
  Use \string\defglsentryfmt\space instead}%
  \expandafter\def\csname gls@#1@display\endcsname##1##2##3##4{#2}%
  \edef\@gls@doentrydef{%
    \noexpand\defglsentryfmt[#1]{%
      \noexpand\ifcsdef{gls@#1@displayfirst}%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\glsdisplayfirst}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
    }%
  }%
  \@gls@doentrydef
}
\newcommand*\defglsdisplayfirst}[2][\glsdefaulttype]{%
  \GlossariesWarning{\string\defglsdisplayfirst\space is now obsolete.^^J
  Use \string\defglsentryfmt\space instead}%
  \expandafter\def\csname gls@#1@displayfirst\endcsname##1##2##3##4{#2}%
  \edef\@gls@doentrydef{%
    \noexpand\defglsentryfmt[#1]{%
      \noexpand\ifcsdef{gls@#1@display}%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
    }%
  }%
  \@gls@doentrydef
}

```

```

    }%
    {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\glsdisplay}%
    }%
} %
} %
\@gls@doentrydef
}
\define@key{glslink}{counter}{%
    \ifcsundef{c@#1}%
    {%
        \PackageError{glossaries}%
        {There is no counter called '#1'}%
        {%
            The counter key should have the name of a valid counter
            as its value%
        }%
    }%
} %
{%
    \def\@gls@counter{#1}%
} %
}
\define@key{glslink}{format}{%
    \def\@glsnumberformat{#1}}
\define@boolkey{glslink}{hyper}[true]{}
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\define@boolkey{glslink}{local}[true]{}
\newcommand*{\glslinkvar}[3]{#1}
\newcommand*{\glsifhyper}[2]{%
    \glslinkvar{#1}{#2}{#1}%
    \GlossariesWarning{\string\glsifhyper\space is deprecated. Did
        you mean \string\glsifhyperon\space or \string\glslinkvar?}%
}
\newcommand*{\@gls@hyp@opt}[1]{%
    \let\glslinkvar\@firstofthree
    \let\@gls@hyp@opt@cs#1\relax
    \@ifstar{\s@gls@hyp@opt}%
    {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*{\s@gls@hyp@opt}[1] []{%
    \let\glslinkvar\@secondofthree
    \@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1] []{%
    \let\glslinkvar\@thirdofthree
    \@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
    \@gls@hyp@opt\@gls@@link
}

```

```

\newcommand*{\@gls@link}[3][\%
  \glsdoifexistsordo{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \@gls@link[#1]{#2}{#3}%
  }{%
    \glstextformat{#3}%
  }%
  \glspostlinkhook
}
\newcommand*{\glspostlinkhook}{}
\newcommand*{\@gls@link@checkfirsthyper}{%
  \ifglsused{\glslabel}%
  {%
  }%
  {%
    \gls@checkisacronymlist\glstype
    \ifglshyperfirst
      \if@glsisacronymlist
        \ifglsacrfootnote
          \KV@glslink@hyperfalse
        \fi
      \fi
    \else
      \KV@glslink@hyperfalse
    \fi
  }%
  \glslinkcheckfirsthyperhook
}
\newcommand*{\glslinkcheckfirsthyperhook}{}
\newcommand*{\glslinkpostsetkeys}{}
\newcommand{\@glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}
\newcommand*{\do@glsdisablehyperinlist}{%
  \expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
  {\KV@glslink@hyperfalse}{}%
}
\newcommand*{\@gls@setdefault@glslink@opts}{}
\def\@gls@link[#1]#2#3{%
  \leavevmode
  \edef\glslabel{\glsdetoklabel{#2}}%
  \def\@gls@link@opts{#1}%
  \let\@gls@link@label\glslabel
  \def\@glsnumberformat{glsnumberformat}%
  \edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  \edef\glstype{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
  \@gls@setdefault@glslink@opts
  \do@glsdisablehyperinlist
  \do@gls@link@checkfirsthyper
  \setkeys{glslink}{#1}%
}

```

```

\glslinkpostsetkeys
\@gls@saveentrycounter
\@gls@setsort{\glslabel}%
\@do@wrglossary{#2}%
\ifKV@glslink@hyper
  \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
\else
  \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
\fi
\let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
}
\newcommand*{\glolinkprefix}{glo:}
\def\glsentrycounter{\glscounter}%
\newcommand*{\@gls@saveentrycounter}{%
  \def\@gls@Hcounter{}%
  \ifthenelse{\equal{\@gls@counter}{equation}}%
  {
    \ifcsundef{xatlevel@}%
    {%
      \edef\theglsentrycounter{\expandafter\noexpand
        \csname the\@gls@counter\endcsname}%
    }%
    {%
      \ifx\xatlevel@\@empty
        \edef\theglsentrycounter{\expandafter\noexpand
          \csname the\@gls@counter\endcsname}%
      \else
        \savecounters@
        \advance\c@equation by 1\relax
        \edef\theglsentrycounter{\csname the\@gls@counter\endcsname}%
        \ifcsundef{theH\@gls@counter}%
        {%
          \def\@gls@Hcounter{\theglsentrycounter}%
        }%
        {%
          \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
        }%
        \protected@edef\theHglsentrycounter{\@gls@Hcounter}%
        \restorecounters@
      \fi
    }%
  }%
  {%
    \edef\theglsentrycounter{\expandafter\noexpand
      \csname the\@gls@counter\endcsname}%
  }%
  \ifx\@gls@Hcounter\@empty
    \ifcsundef{theH\@gls@counter}%
    {%
      \def\theHglsentrycounter{\theglsentrycounter}%
    }%
  }%

```

```

}%
{%
\protected@edef\theHglentrycounter{\expandafter\noexpand
\csname theH\@gls@counter\endcsname}%
}%
\fi
}
\def\@set@glo@numformat#1#2#3#4{%
\expandafter\@glo@check@mkidxrangear#3\@nil
\protected@edef#1{%
\@glo@prefix setentrycounter[#4]{#2}%
\expandafter\string\csname\@glo@suffix\endcsname
}%
\@gls@checkmkidxchars#1%
}
\def\@glo@check@mkidxrangear#1#2\@nil{%
\if#1\relax
\def\@glo@prefix{()}%
\if\relax#2\relax
\def\@glo@suffix{glsnumberformat}%
\else
\def\@glo@suffix{#2}%
\fi
\else
\if#1\relax
\def\@glo@prefix{}%
\if\relax#2\relax
\def\@glo@suffix{glsnumberformat}%
\else
\def\@glo@suffix{#2}%
\fi
\else
\def\@glo@prefix{\def\@glo@suffix{#1#2}}%
\fi
\fi}
\newcommand*{\@gls@escbsdq}[1]{%
\def\@gls@checkedmkidx{}%
\let\@gls@xdystring=#1\relax
\@onelevel@sanitize\@gls@xdystring
\edef\do@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash\@gls@xdystring\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null}%
\do@gls@xdycheckbackslash
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\@gls@xdystring}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@xdycheckquote\@gls@xdystring\@nil""\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\@gls@xdystring}%
\@for\@gls@tmp:=\@gls@protected@pagefmts\do
{%
\edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\expandonce\@gls@tmp}%

```

```

    \@onelevel@sanitize\@gls@sanitized@tmp
    \edef\gls@dostsubst{%
      \noexpand\DTLsubstituteall\noexpand\gls@xdystring
      {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
    }%
    \gls@dostsubst
  }%
  \let#1=\gls@xdystring
}
\newcommand{\@gls@checkmkidxchars}[1]{%
  \ifglsxindy
    \@gls@escbsdq{#1}%
  \else
    \def\@gls@checkedmkidx{%
      \expandafter\@gls@checkquote#1\@nil""\null
      \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
      \def\@gls@checkedmkidx{%
        \expandafter\@gls@checkescquote#1\@nil\\"\null
        \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
        \def\@gls@checkedmkidx{%
          \expandafter\@gls@checkescactual#1\@nil\?\?\null
          \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
          \def\@gls@checkedmkidx{%
            \expandafter\@gls@checkactual#1\@nil??\null
            \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
            \def\@gls@checkedmkidx{%
              \expandafter\@gls@checkbar#1\@nil||\null
              \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
              \def\@gls@checkedmkidx{%
                \expandafter\@gls@checkescbar#1\@nil|||\null
                \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                \def\@gls@checkedmkidx{%
                  \expandafter\@gls@checklevel#1\@nil!!\null
                  \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                }%
              }%
            }%
          }%
        }%
      }%
    }%
  \fi
}
\def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
\newtoks\@gls@tmpb
\def\@gls@checkquote#1"#2"#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\@nil#2\@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\null}%
    \fi
  \fi
}

```

```

\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@quotechar}%
\ifx\null#3\null
\def\@gls@checkquote{\@gls@checkquote#2""\null}%
\else
\def\@gls@checkquote{\@gls@checkquote#2"#3\null}%
\fi
\fi
\@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@

```



```

\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{%

```



```

}%
{%
  \let\@glslink\glsdohyperlink
}
\ifcsundef{hypertarget}%
{%
  \let\@glstarget\@secondoftwo
}%
{%
  \let\@glstarget\glsdohypertarget
}
\newcommand{\glsdisablehyper}{%
  \KV@glslink@hyperfalse
  \let\@glslink\glsdonohyperlink
  \let\@glstarget\@secondoftwo
}
\newcommand{\glsenablehyper}{%
  \KV@glslink@hypertrue
  \let\@glslink\glsdohyperlink
  \let\@glstarget\glsdohypertarget
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}
\newcommand*{\@gls}[2][]{%
  \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2} []}%
}
\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glsescapscase\@firstofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
\newcommand*{\@Gls}[2][]{%
  \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2} []}%
}
\def\@Gls@#1#2[#3]{%

```

```

\glsdoifexists{#2}%
{%
  \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
  \let\glsifplural\@secondoftwo
  \let\glscapscase\@secondofthree
  \let\glscustomtext\@empty
  \def\glsinsert{#3}%
  \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
  \@gl@link[#1]{#2}{\@glo@text}%
  \ifKV@gl@link@local
    \glslocalunset{#2}%
  \else
    \glsunset{#2}%
  \fi
}%
\glspostlinkhook
}
\newrobustcmd*{\GLS}{\@gl@hyp@opt\@GLS}
\newcommand*{\@GLS}[2] []{%
  \new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2} []}%
}
\def\@GLS@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \ifKV@gl@link@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\glspl}{\@gl@hyp@opt\@glspl}
\newcommand*{\@glspl}[2] []{%
  \new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2} []}%
}
\def\@glspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glscustomtext\@empty

```

```

\def\glsinsert{#3}%
\def\glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\ifKV@glslink@local
  \glslocalunset{#2}%
\else
  \glsunset{#2}%
\fi
}%
\glspostlinkhook
}
\newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
\newcommand*{\@Glspl}[2][{}]{%
  \new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2}[]}%
}
\def\@Glspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\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*{\GLSp1}{\@gls@hyp@opt\@GLSp1}
\newcommand*{\@GLSp1}[2][{}]{%
  \new@ifnextchar[{\@GLSp1@{#1}{#2}}{\@GLSp1@{#1}{#2}[]}%
}
\def\@GLSp1@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glsapscase\@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\glsescapscase\@firstofthree
        \def\glscustomtext{#3}%
        \def\glsinsert{}%
        \def\glo@text{\csname gls@glstype @entryfmt\endcsname}%
        \@gls@link{#1}{#2}{\@glo@text}%
        \ifKV@glslink@local
            \glslocalunset{#2}%
        \else
            \glsunset{#2}%
        \fi
    }%
    \glspostlinkhook
}
\newcommand*{\@gls@link@nocheckfirsthyper}{}
\newcommand{\@gls@field@link}[3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \@gls@link{#1}{#2}{#3}%
    }%
    \glspostlinkhook
}
\newrobustcmd*{\glsstext}{\@gls@hyp@opt\@glsstext}
\newcommand*{\@glsstext}[2] []{%
    \new@ifnextchar[{\@glsstext@{#1}{#2}}{\@glsstext@{#1}{#2} []}]
\def\@glsstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentrytext{#2}#3}%
}
\newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
\newcommand*{\@GLStext}[2] []{%
    \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2} []}]
\def\@GLStext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrytext{#2}#3}}%
}
\newrobustcmd*{\Glstext}{\@gls@hyp@opt\@Glstext}
\newcommand*{\@Glstext}[2] []{%
    \new@ifnextchar[{\@Glstext@{#1}{#2}}{\@Glstext@{#1}{#2} []}]
\def\@Glstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentrytext{#2}#3}%
}
}

```

```

\newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
\newcommand*{\@glsfirst}[2] [] {%
  \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2} []}]
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
}
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
\newcommand*{\@Glsfirst}[2] [] {%
  \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2} []}]
\def\@Glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
\newcommand*{\@GLSfirst}[2] [] {%
  \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2} []}]
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirst{#2}#3}}%
}
\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
\newcommand*{\@glsplural}[2] [] {%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} []}]
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\newcommand*{\@Glsplural}[2] [] {%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} []}]
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
\newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
\newcommand*{\@GLSplural}[2] [] {%
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} []}]
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryplural{#2}#3}}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{\@glsfirstplural}[2] [] {%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} []}]
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
\newcommand*{\@Glsfirstplural}[2] [] {%
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2} []}]
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
\newcommand*{\@GLSfirstplural}[2] [] {%

```



```

\new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} []}]
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirstplural{#2}#3}}%
}
\newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
\newcommand*{\@glsname}[2] []{%
  \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2} []}]
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryname{#2}#3}%
}
\newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}
\newcommand*{\@Glsname}[2] []{%
  \new@ifnextchar[{\@Glsname@{#1}{#2}}{\@Glsname@{#1}{#2} []}]
\def\@Glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryname{#2}#3}%
}
\newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}
\newcommand*{\@GLSname}[2] []{%
  \new@ifnextchar[{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}]
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryname{#2}#3}}%
}
\newrobustcmd*{\glsdesc}{\@gls@hyp@opt\@glsdesc}
\newcommand*{\@glsdesc}[2] []{%
  \new@ifnextchar[{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2} []}]
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydesc{#2}#3}%
}
\newrobustcmd*{\Glsdesc}{\@gls@hyp@opt\@Glsdesc}
\newcommand*{\@Glsdesc}[2] []{%
  \new@ifnextchar[{\@Glsdesc@{#1}{#2}}{\@Glsdesc@{#1}{#2} []}]
\def\@Glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%
}
\newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\newcommand*{\@GLSdesc}[2] []{%
  \new@ifnextchar[{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}]
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydesc{#2}#3}}%
}
\newrobustcmd*{\glsdescplural}{\@gls@hyp@opt\@glsdescplural}
\newcommand*{\@glsdescplural}[2] []{%
  \new@ifnextchar[{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}]
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\Glsdescplural}{\@gls@hyp@opt\@Glsdescplural}
\newcommand*{\@Glsdescplural}[2] []{%
  \new@ifnextchar[{\@Glsdescplural@{#1}{#2}}{\@Glsdescplural@{#1}{#2} []}]
\def\@Glsdescplural@#1#2[#3]{%

```

```

    \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\newcommand*{\@GLSdescplural}[2] [] {%
    \new@ifnextchar [{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}]
\def \@GLSdescplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydescplural{#2}#3}}%
}
\newrobustcmd*{\glssymbol}{\@gls@hyp@opt\@glssymbol}
\newcommand*{\@glssymbol}[2] [] {%
    \new@ifnextchar [{\@glssymbol@{#1}{#2}}{\@glssymbol@{#1}{#2} []}]
\def \@glssymbol@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\Glsymbol}{\@gls@hyp@opt\@Glsymbol}
\newcommand*{\@Glsymbol}[2] [] {%
    \new@ifnextchar [{\@Glsymbol@{#1}{#2}}{\@Glsymbol@{#1}{#2} []}]
\def \@Glsymbol@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\Glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\GLSsymbol}{\@gls@hyp@opt\@GLSsymbol}
\newcommand*{\@GLSsymbol}[2] [] {%
    \new@ifnextchar [{\@GLSsymbol@{#1}{#2}}{\@GLSsymbol@{#1}{#2} []}]
\def \@GLSsymbol@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrysymbol{#2}#3}}%
}
\newrobustcmd*{\glssymbolplural}{\@gls@hyp@opt\@glssymbolplural}
\newcommand*{\@glssymbolplural}[2] [] {%
    \new@ifnextchar [{\@glssymbolplural@{#1}{#2}}{\@glssymbolplural@{#1}{#2} []}]
\def \@glssymbolplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\Glsymbolplural}{\@gls@hyp@opt\@Glsymbolplural}
\newcommand*{\@Glsymbolplural}[2] [] {%
    \new@ifnextchar [{\@Glsymbolplural@{#1}{#2}}{\@Glsymbolplural@{#1}{#2} []}]
\def \@Glsymbolplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrysymbolplural{#2}#3}}%
}
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
\newcommand*{\@glsuseri}[2] [] {%
    \new@ifnextchar [{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2} []}]
\def \@glsuseri@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsentryuseri{#2}#3}%
}
}

```

```

\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
\newcommand*{\@Glsuseri}[2][{}]{%
  \new@ifnextchar[{\@Glsuseri@#1}{#2}]{\@Glsuseri@{#1}{#2}[{}]}
\def\@Glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\newcommand*{\@GLSuseri}[2][{}]{%
  \new@ifnextchar[{\@GLSuseri@{#1}{#2}]{\@GLSuseri@{#1}{#2}[{}]}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseri{#2}#3}}%
}
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
\newcommand*{\@glsuserii}[2][{}]{%
  \new@ifnextchar[{\@glsuserii@{#1}{#2}]{\@glsuserii@{#1}{#2}[{}]}
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
\newcommand*{\@Glsuserii}[2][{}]{%
  \new@ifnextchar[{\@Glsuserii@{#1}{#2}]{\@Glsuserii@{#1}{#2}[{}]}
\def\@Glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\GLSuserii}{\@gls@hyp@opt\@GLSuserii}
\newcommand*{\@GLSuserii}[2][{}]{%
  \new@ifnextchar[{\@GLSuserii@{#1}{#2}]{\@GLSuserii@{#1}{#2}[{}]}
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuserii{#2}#3}}%
}
\newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}
\newcommand*{\@glsuseriii}[2][{}]{%
  \new@ifnextchar[{\@glsuseriii@{#1}{#2}]{\@glsuseriii@{#1}{#2}[{}]}
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
\newcommand*{\@Glsuseriii}[2][{}]{%
  \new@ifnextchar[{\@Glsuseriii@{#1}{#2}]{\@Glsuseriii@{#1}{#2}[{}]}
\def\@Glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseriii{#2}#3}}%
}
\newrobustcmd*{\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@\gls@type @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\glsapsaps\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls@type @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshort}{\@gls@hyp@opt\ns@ACRshort}
\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\glsapsaps\@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\@CRshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{#2}}#3}%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*\@acrlong{\@gl@hyp@opt\@ns@acrlong}
\newcommand*\@ns@acrlong}[2][{}]{%
  \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2}[]}%
}
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsentrylong{#2}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*\@Acrlong{\@gl@hyp@opt\@ns@Acrlong}
\newcommand*\@ns@Acrlong}[2][{}]{%
  \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2}[]}%
}
\def\@Acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylong{#2}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%

```

```

}%
\glspostlinkhook
}
\newrobustcmd*{\ACRlong}{\@gls@hyp@opt\ns@ACRlong}
\newcommand*{\ns@ACRlong}[2] [] {%
  \new@ifnextchar [{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2} []}%
}
\def\@ACRlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\glsentrylong{#2}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\ns@acrlongpl}
\newcommand*{\ns@acrlongpl}[2] [] {%
  \new@ifnextchar [{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2} []}%
}
\def\@acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\ns@Acrlongpl}
\newcommand*{\ns@Acrlongpl}[2] [] {%
  \new@ifnextchar [{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2} []}%
}
\def\@Acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%

```



```

\let\glsifplural\@firstoftwo
\let\glscapscase\@secondofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \Glsentrylongpl{#2}#3%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\@ns@ACRlongpl}
\newcommand*{\ns@ACRlongpl}[2][{}]{%
  \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2}[]}%
}
\def\@ACRlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glsapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\glsentrylongpl{#2}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newcommand*{\@gls@entry@field}[2]{%
  \csname glo@\glsdetoklabel{#1}@#2\endcsname
}
\newcommand*{\glsletentryfield}[3]{%
  \letcs{#1}{glo@\glsdetoklabel{#2}@#3}%
}
\newcommand*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@#2}%
    \ifdef\@glo@text
    {%
      \xmakefirstuc{\@glo@text}%
    }%
  }%
  ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
  entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
  label and the field name}%
}%
}%

```

```

    ???%
  }%
}
\newcommand*\glsentryname}[1]{\@gls@entry@field{#1}{name}}
\newrobustcmd*\Glsentryname}[1]{%
  \@Gls@entryname{#1}%
}
\newcommand*\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
\newcommand*\@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo\@glsdetoklabel{#1}@name}%
    \expandafter\@gls@getbody\@glo@text{}\@nil
    \expandafter\ifx\@gls@body\glsentrylong\relax
    \expandafter\Glsentrylong\@gls@rest
  \else
    \expandafter\ifx\@gls@body\glsentryshort\relax
    \expandafter\Glsentryshort\@gls@rest
  \else
    \expandafter\ifx\@gls@body\acronymfont\relax
    {%
      \let\glsentryshort\Glsentryshort
      \@glo@text
    }%
  \else
    \expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
    {%
      \let\glsentryshort\Glsentryshort
      \@glo@text
    }%
  \else
    \xmakefirstuc{\@glo@text}%
  \fi
  \fi
  \fi
  }%
  }%
  \@Gls@entry@field{#1}{name}%
}%
}
\newcommand*\glsentrydesc}[1]{\@gls@entry@field{#1}{desc}}
\newrobustcmd*\Glsentrydesc}[1]{%
  \@Gls@entry@field{#1}{desc}%
}
\newcommand*\glsentrydescplural}[1]{%
  \@gls@entry@field{#1}{descplural}%
}
}

```

```

\newrobustcmd*\Glsentrydescplural}[1]{%
  \@Gls@entry@field{#1}{descplural}%
}
\newcommand*\glsentrytext}[1]{\@gls@entry@field{#1}{text}}
\newrobustcmd*\Glsentrytext}[1]{%
  \@Gls@entry@field{#1}{text}%
}
\newcommand*\glsentryplural}[1]{%
  \@gls@entry@field{#1}{plural}%
}
\newrobustcmd*\Glsentryplural}[1]{%
  \@Gls@entry@field{#1}{plural}%
}
\newcommand*\glsentrysymbol}[1]{%
  \@gls@entry@field{#1}{symbol}%
}
\newrobustcmd*\Glsentrysymbol}[1]{%
  \@Gls@entry@field{#1}{symbol}%
}
\newcommand*\glsentrysymbolplural}[1]{%
  \@gls@entry@field{#1}{symbolplural}%
}
\newrobustcmd*\Glsentrysymbolplural}[1]{%
  \@Gls@entry@field{#1}{symbolplural}%
}
\newcommand*\glsentryfirst}[1]{%
  \@gls@entry@field{#1}{first}%
}
\newrobustcmd*\Glsentryfirst}[1]{%
  \@Gls@entry@field{#1}{first}%
}
\newcommand*\glsentryfirstplural}[1]{%
  \@gls@entry@field{#1}{firstpl}%
}
\newrobustcmd*\Glsentryfirstplural}[1]{%
  \@Gls@entry@field{#1}{firstpl}%
}
\newrobustcmd*\@glsentrytitlecase}[2]{%
  \glsdoifexists{#1}%
  {%
    \glsfieldfetch{#1}{#2}{\@gls@value}%
    \xcapitalisewords{\@gls@value}%
  }%
}
\ifdef\texorpdfstring
{
\newcommand*\glsentrytitlecase}[2]{%
  \texorpdfstring
  {\@glsentrytitlecase{#1}{#2}}%
  {\@gls@entry@field{#1}{#2}}%
}

```

```

    }
  }
  {
    \newcommand*{\glsentrytitlecase}[2]{\@glsentrytitlecase{#1}{#2}}
  }
  \newcommand*{\glsentrytype}[1]{\@gls@entry@field{#1}{type}}
  \newcommand*{\glsentrysort}[1]{%
    \@gls@entry@field{#1}{sort}%
  }
  \newcommand*{\glsentryparent}[1]{%
    \@gls@entry@field{#1}{parent}%
  }
  \newcommand*{\glsentryuseri}[1]{%
    \@gls@entry@field{#1}{useri}%
  }
  \newrobustcmd*{\Glsentryuseri}[1]{%
    \@Gls@entry@field{#1}{useri}%
  }
  \newcommand*{\glsentryuserii}[1]{%
    \@gls@entry@field{#1}{userii}%
  }
  \newrobustcmd*{\Glsentryuserii}[1]{%
    \@Gls@entry@field{#1}{userii}%
  }
  \newcommand*{\glsentryuseriii}[1]{%
    \@gls@entry@field{#1}{useriii}%
  }
  \newrobustcmd*{\Glsentryuseriii}[1]{%
    \@Gls@entry@field{#1}{useriii}%
  }
  \newcommand*{\glsentryuseriv}[1]{%
    \@gls@entry@field{#1}{useriv}%
  }
  \newrobustcmd*{\Glsentryuseriv}[1]{%
    \@Gls@entry@field{#1}{useriv}%
  }
  \newcommand*{\glsentryuserv}[1]{%
    \@gls@entry@field{#1}{userv}%
  }
  \newrobustcmd*{\Glsentryuserv}[1]{%
    \@Gls@entry@field{#1}{userv}%
  }
  \newcommand*{\glsentryuservi}[1]{%
    \@gls@entry@field{#1}{uservi}%
  }
  \newrobustcmd*{\Glsentryuservi}[1]{%
    \@Gls@entry@field{#1}{uservi}%
  }
  \newcommand*{\glsentryshort}[1]{\@gls@entry@field{#1}{short}}
  \newrobustcmd*{\Glsentryshort}[1]{%

```

```

    \@Gls@entry@field{#1}{short}%
  }
  \newcommand*{\glsentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}
  \newrobustcmd*{\Glsentryshortpl}[1]{%
    \@Gls@entry@field{#1}{shortpl}%
  }
  \newcommand*{\glsentrylong}[1]{\@gls@entry@field{#1}{long}}
  \newrobustcmd*{\Glsentrylong}[1]{%
    \@Gls@entry@field{#1}{long}%
  }
  \newcommand*{\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}
  \newrobustcmd*{\Glsentrylongpl}[1]{%
    \@Gls@entry@field{#1}{longpl}%
  }
  \newcommand*{\glsentryfull}[1]{%
    \acrfullformat{\glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
  }
  \newrobustcmd*{\Glsentryfull}[1]{%
    \acrfullformat{\Glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
  }
  \newcommand*{\glsentryfullpl}[1]{%
    \acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
  }
  \newrobustcmd*{\Glsentryfullpl}[1]{%
    \acrfullformat{\Glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
  }
  \newcommand*{\glsentrynumberlist}[1]{%
    \glsdoifexists{#1}%
    {%
      \@gls@entry@field{#1}{numberlist}%
    }%
  }
  \ifpackageloaded{hyperref} {%
    \newcommand*{\glsdisplaynumberlist}[1]{%
      \GlossariesWarning
      {%
        \string\glsdisplaynumberlist\space
        doesn't work with hyperref.^^JUsing
        \string\glsentrynumberlist\space instead%
      }%
      \glsentrynumberlist{#1}%
    }%
  }%
  {%
    \newcommand*{\glsdisplaynumberlist}[1]{%
      \glsdoifexists{#1}%
      {%
        \bgroup
        \edef\@glo@label{\glsdetoklabel{#1}}%
        \let\@org@glsnumberformat\glsnumberformat

```

```

\def\glsnumberformat##1{##1}%
\protected@edef\the@numberlist{%
  \csname glo@\@glo@label @numberlist\endcsname}%
\def\@gls@numlist@sep{}%
\def\@gls@numlist@nextsep{}%
\def\@gls@numlist@lastsep{}%
\def\@gls@thislist{}%
\def\@gls@donext@def{}%
\renewcommand\do[1]{%
  \protected@edef\@gls@thislist{%
    \@gls@thislist
    \noexpand\@gls@numlist@sep
    ##1%
  }%
  \let\@gls@numlist@sep\@gls@numlist@nextsep
  \def\@gls@numlist@nextsep{\glsnumlistsep}%
  \@gls@donext@def
  \def\@gls@donext@def{%
    \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
  }%
}%
\expandafter \glsnumlistparser \expandafter{\the@numberlist}%
\let\@gls@numlist@sep\@gls@numlist@lastsep
\@gls@thislist
\egroup
}%
}
}
\newcommand*\glsnumlistsep}{, }
\newcommand*\glsnumlistlastsep}{ \& }
\newcommand*\gls hyperlink}[2][\glsentrytext{\@glo@label}]{%
  \def\@glo@label{#2}%
  \@glslink{\glo@linkprefix\glsdetoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*\glsadd}[2][{}]{%
  \@gls@adjustmode
  \glsdoifexists{#2}%
  {%
    \def\@glsnumberformat{\glsnumberformat}%
    \edef\@gls@counter{\csname glo@\glsdetoklabel{#2}@counter\endcsname}%
    \setkeys{glossadd}{#1}%
    \@gls@saveentrycounter
    \@gls@setsort{#2}%
    \@do@wrglossary{#2}%
  }%
}
\newcommand*\@gls@adjustmode}{
\AtBeginDocument{\renewcommand*\@gls@adjustmode}{\ifvmode\mbox{}\fi}}

```

```

\newrobustcmd*\glsaddall}[1] [] {%
  \edef\glo@type{\@glo@types}%
  \setkeys{glossadd}{#1}%
  \forallglsentries[\@glo@type]{\@glo@entry}{%
    \glsadd[#1]{\@glo@entry}%
  }%
}
\newrobustcmd*\glsaddallunused}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glo@entry}%
  {%
    \ifglsused{\@glo@entry}{\glsadd[format=glsignore]{\@glo@entry}}%
  }%
}
\newcommand*\glsignore}[1]{}
\edef\glsopenbrace{\expandafter\@gobble\string\{ }
\edef\glsclosebrace{\expandafter\@gobble\string\} }
\edef\glsbackslash{\expandafter\@gobble\string\ }
\edef\glsquote#1{\string"#1\string"}
\edef\glspercentchar{\expandafter\@gobble\string\% }
\edef\glsstildechar{\string~ }
\ifglsxindy
  \newcommand*\@glsfirstletter}{A}
\fi
\newcommand*\GlsSetXdyFirstLetterAfterDigits){%
  \@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
\ifglsxindy
  \newcommand*\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*\@glsfirstletter}{#1}}
  \newcommand*\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*\@glsfirstletter}{#1}%
    \@onelevel@sanitize\@glsfirstletter
  }
\else
  \newcommand*\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \glsnoxywarning\GlsSetXdyFirstLetterAfterDigits}
  \newcommand*\s@GlsSetXdyFirstLetterAfterDigits){%
    \@GlsSetXdyFirstLetterAfterDigits
  }
\fi
\ifglsxindy
  \newcommand*\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
\fi
\newcommand*\GlsSetXdyNumberGroupOrder}[1]{%
  \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
}
\ifglsxindy
  \newcommand*\@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*\@xdynumbergrouporder}{#1}%
  }
  \newcommand*\s@GlsSetXdyNumberGroupOrder}[1]{%

```

```

\renewcommand*\@xdynumbergrouporder}{#1}%
\@onelevel@sanitize\@xdynumbergrouporder
}
\else
\newcommand*\@GlsSetXdyNumberGroupOrder}[1]{%
\glsnoxywarning\GlsSetXdyNumberGroupOrder}
\newcommand*\@s@GlsSetXdyNumberGroupOrder}{%
\@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*\@glsminrange}{2}
\ifglsxindy
\newcommand*\@GlsSetXdyMinRangeLength}[1]{%
\renewcommand*\@glsminrange}{#1}}
\else
\newcommand*\@GlsSetXdyMinRangeLength}[1]{%
\glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglsxindy
\def\writeist{%
\ifundef\glswrite{\newwrite\glswrite}{}%
\@gls@addpredefinedattributes
\openout\glswrite=\istfilename
\write\glswrite{;; xindy style file created by the glossaries
package}%
\write\glswrite{;; for document '\jobname' on
\the\year-\the\month-\the\day}%
\write\glswrite{^^J; required styles^^J}
\@for\@xdystyle:=\@xdyrequiredstyles\do{%
\ifx\@xdystyle\@empty
\else
\protected@write\glswrite}{(require
\string"\@xdystyle.xdy\string")}%
\fi
}%
\write\glswrite{^^J%
; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%
\@for\@gls@classI:=\@gls@xdy@locationlist\do{%
\protected@write\glswrite}{(define-location-class
\string"\@gls@classI\string"^^J\space\space\space
(
:sep "{ }{"
\csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
:sep "}"
)
^^J\space\space\space

```



```

: min-range-length \@glsminrange^^J%
)
}%
{%
\@for\@gls@classII:=\@gls@xdy@locationlist\do{%
  \protected@write\glswrite{}\{(define-location-class
    \string"\@gls@classII-\@gls@classI\string"
    ^^J\space\space\space
    (
      :sep "{"
      \csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
      :sep "}"
      \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
      :sep "}"
    )
    ^^J\space\space\space
    : min-range-length \@glsminrange^^J%
  )
}%
}%
}%
}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\glsseeformat\string"
  :close \string"{}\string")}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
  :open \string"\string
  \glossarysection[\string\glossarytoctitle]{\string
  \glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
  {%
    \@for\@this@attr:=\@xdyattributelist\do{%
      \protected@write\glswrite{}\{\string\providecommand*
        \expandafter\string
        \csname glsX\@this@ctr X\@this@attr\endcsname[2]%
        {%
          \string\setentrycounter
            [\expandafter\@gobble\string\#1]{\@this@ctr}%
          \expandafter\string

```

```

\csname\@this@attr\endcsname
  {\expandafter\@gobble\string\#2}%
}%
}%
}%
}%
\write\glswrite{%
  \string\begin
  {theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
  \space\space:close \string" \glpercentchar\glstildechar n\string
  \end{theglossary}\string\glossarypostamble
  \glstildechar n\string" ^^J\space\space\space
  :tree)}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \glstildechar n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list
  :sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
  :sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
  :open-head \string"\string\glsgroupheading
  \glsopenbrace\string"^^J\space\space\space
  :close-head \string"\glsclosebrace\string")}%

```

```

\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}%
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax
\ifundef{\glswrite}{\newwrite\glswrite}{}\relax
\openout\glswrite=\istfilename
\write\glswrite{\glspercentchar\space makeindex style file
created by the glossaries package}
\write\glswrite{\glspercentchar\space for document
'\jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\@gls@actualchar'}
\write\glswrite{encap '\@gls@encapchar'}
\write\glswrite{level '\@gls@levelchar'}
\write\glswrite{quote '\@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
\glossarytoctitle]{\string\glossarytitle}\string
\glossarypreamble\string\n\string\begin{theglossary}\string
\glossaryheader\string\n\string"}
\write\glswrite{postamble \string"\string%\string\n\string
\end{theglossary}\string\glossarypostamble\string\n
\string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
\string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}
\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{delim_0 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string\relax \string"}
\glossaryentrynumbers\string\{\string\relax \string"}

```

```

\write\glswrite{delim_2 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string}
\write\glswrite{delim_t \string"\string}\string}\string"}
\write\glswrite{delim_n \string"\string\delimN \string"}
\write\glswrite{delim_r \string"\string\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
  \string"\string}\string\relax
  \string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"glscpositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\fi
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{
\ifglxsindy
\newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\newcommand*\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}%
\@ifpackageloaded{tracklang}%
{%
\IfTrackedLanguage{german}%
{%
\def\@gls@extramakeindexopts{-g}%
}%
}%
}%
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkquote####1#1####2#1####3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={###1}%

```

```

\noexpand\ifx\noexpand\null###2\noexpand\null
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks}%
\noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###2#1#1\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###2#1###3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkquote###1\noexpand\@nil
#1#1\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkesquote###1\noexpand\@nil
\expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkesactual###1\noexpand\@nil
\noexpand\?\noexpand\?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkactual###1\noexpand\@nil
\noexpand?\noexpand?\noexpand\null

```

```

\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkbar###1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkesobar###1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checklevel###1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkesquote###1%
\expandonce{\csname#1\endcsname}###2\expandonce{\csname#1\endcsname}%
###3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={###1}%
\noexpand\ifx\noexpand\null###2\noexpand\null
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkesquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkesquote{%
\noexpand\@gls@checkesquote###3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string
\expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@gls@checkesquote{%
\noexpand\@gls@checkesquote###2\expandonce{\csname#1\endcsname}%
\expandonce{\csname#1\endcsname}\noexpand\null}%

```

```

        \noexpand\else
        \noexpand\def\noexpand\@gls@checkescquote{%
            \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
            ####3\noexpand\null}%
        \noexpand\fi
        \noexpand\fi
        \noexpand\@gls@checkescquote
    }%
}
\@gls@docheckquotedef
}
\newcommand*\@gls@nosetquote}[1]{\PackageError{glossaries}%
    {\string\GlsSetQuote\space not permitted here}%
    {Move \string\GlsSetQuote\space earlier in the preamble, as
    soon as possible after glossaries.sty has been loaded}}
\fi
\newcommand*\@gls@extramakeindexopts}[1]{
\newcommand*\@noist}{%
    \@gls@addpredefinedattributes
    \let\writeist\relax
}
\newcommand*\@makeglossary}[1]{%
    \ifglossaryexists{#1}%
    {%
        \ifglssavewrites
            \expandafter\newtoks\csname glo@#1@filetok\endcsname
        \else
            \expandafter\newwrite\csname glo@#1@file\endcsname
            \expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
        \fi
        \@gls@renewglossary
        \writeist
    }%
    {%
        \PackageError{glossaries}%
        {Glossary type ‘#1’ not defined}%
        {New glossaries must be defined before using \string\makeglossaries}%
    }%
}
\newcommand*\@glsopenfile}[2]{%
    \immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
    \PackageInfo{glossaries}{Writing glossary file
        \jobname.\csname @glotype@#2@out\endcsname}%
}
\newcommand*\@closegls}[1]{%
    \closeout\csname glo@#1@file\endcsname
}
\ifglsexindy
\newcommand*\@gls@automake}[1]{%
    \ifglossaryexists{#1}

```

```

{%
  \@closegls{#1}%
  \ifdefstring{\glsorder}{letter}%
  {\def\@gls@order{-M ord/letorder }}%
  {\let\@gls@order\@empty}%
  \ifcsundef{@xdy@#1@language}%
  {\let\@gls@langmod\@xdy@main@language}%
  {\letcs\@gls@langmod{@xdy@#1@language}}%
  \edef\@gls@dothiswrite{\noexpand\write18{xindy
    -I xindy
    \@gls@order
    -L \@gls@langmod\space
    -M \@gls@istfilebase\space
    -C \@gls@codepage\space
    -t \jobname.\csuse{@glotype@#1@log}
    -o \jobname.\csuse{@glotype@#1@in}
    \jobname.\csuse{@glotype@#1@out}}}%
  }%
  \@gls@dothiswrite
}%
{%
  \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\else
\newcommand*{\@gls@automake}[1]{%
  \ifglossaryexists{#1}
  {%
    \@closegls{#1}%
    \ifdefstring{\glsorder}{letter}%
    {\def\@gls@order{-l }}%
    {\let\@gls@order\@empty}%
    \edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
      -s \istfilename\space
      -t \jobname.\csuse{@glotype@#1@log}
      -o \jobname.\csuse{@glotype@#1@in}
      \jobname.\csuse{@glotype@#1@out}}}%
    }%
    \@gls@dothiswrite
  }%
  {%
    \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
  }%
}
\fi
\ifglsxindy
\newcommand*{\@gls@automake@immediate}[1]{%
  \ifglossaryexists{#1}
  {%
    \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%

```



```

{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-M ord/letorder }}%
{\let\@gls@order\@empty}%
\ifcsundef{@xdy@#1@language}%
{\let\@gls@langmod\@xdy@main@language}%
{\letcs\@gls@langmod{@xdy@#1@language}}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \gls@istfilename\space
-C \gls@codepage\space
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{\GlossariesWarning{Can't make glossary '#1', it doesn't exist}}%
}%
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%
\IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-l }}%
{\let\@gls@order\@empty}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order
-s \istfilename\space
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{\GlossariesWarning{Can't make glossary '#1', it doesn't exist}}%
}%
}
}

```

```

\fi
\newcommand*{\@warn@nomakeglossaries}{}
\newcommand*{\warn@nomakeglossaries}{\@warn@nomakeglossaries}
\newcommand{\@gls@automake@immediate}{%
\ifnum\gls@automake@nr=2\relax
\@for\@gls@type:=\@glo@types\do{%
\ifdefempty{\@gls@type}{}%
{\@gls@automake@immediate{\@gls@type}}%
}%
\glsautomakefalse
\renewcommand*{\@gls@doautomake}{}%
\fi
}
\newcommand*{\makeglossaries}{%
\@domakeglossaries
{%
\@gls@@automake@immediate
\ifundef{\glswrite}{\newwrite\glswrite}{}%
\protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{} }
\protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{} }
\ifundef\@@gls@extramakeindexopts
{}%
{%
\protected@write\@auxout{}{\string\providecommand
\string\@gls@extramakeindexopts[1]{} }
\protected@write\@auxout{}{\string\@gls@extramakeindexopts
{\@@gls@extramakeindexopts}}%
}%
\protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
\protected@write\@auxout{}{\string\@glsorder{\glsorder}}
\@for\@glo@type:=\@glo@types\do{%
\ifthenelse{\equal{\@glo@type}{} }{}{}%
\@makeglossary{\@glo@type}}%
}%
\renewcommand*\newglossary[4] []{%
\PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%
\let\@makeglossary\@gobble
\let\makeglossaries\relax
\@disable@onlypremakeg
\let\gls@checkseeallowed\relax
\let\warn@nomakeglossaries\relax
\def\warn@noprintglossary{%
\ifdefstring{\@glo@types}{,}%
{%
\GlossariesWarningNoLine{No glossaries have been defined}%
}%
}%

```

```

    \GlossariesWarningNoLine{No \string\printglossary\space
      or \string\printglossaries\space
      found. ^^J(Remove \string\makeglossaries\space if you
      don't want any glossaries.) ^^JThis document will not
      have a glossary}%
  }%
}%
\ifglssavenumberlist
  \edef\@gls@dodolistparser{\noexpand\DeclareListParser
    {\noexpand\glsnumlistparser}{\delimN}}%
  \@gls@dodolistparser
\fi
\let\makenoidxglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
  \let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
\ifglsautomake
  \renewcommand*{\@gls@doautomake}{%
    \for\@gls@type:=\@glo@types\do{%
      \ifdefempty{\@gls@type}{%
        {\@gls@automake{\@gls@type}}%
      }%
    }%
  }%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}
\AtEndDocument{%
  \warn@nomakeglossaries
  \warn@noprintglossary
}
\newcommand*{\makenoidxglossaries}{%
  \@domakeglossaries
  {%
    \renewcommand{\@gls@nofwarn}[1]{%
      \GlossariesWarning{Empty glossary for
        \string\printnoidxglossary[type={##1}].
        Rerun may be required (or you may have forgotten to use
        commands like \string\gls)}%
    }%
    \let\@gls@checkmkidxchars\@gobble
    \glsesclocationsfalse
    \let\@do@wrglossary\gls@noidxglossary
    \let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
    \let\gls@checkseeallowed\relax
    \renewcommand{\@do@seeglossary}[2]{%
      \edef\@gls@label{\glsdetoklabel{##1}}%
      \protected@write\@auxout{}{%
        \string\@gls@reference

```

```

        {\csname glo@\@gls@label @type\endcsname}%
        {\@gls@label}%
        {%
        \string\glsseeformat##2{}}%
    }%
}%
}%
\AtBeginDocument
{%
\write\@auxout{\string\providecommand\string\@gls@reference[3]{}%
}%
\def\warn@noprintglossary{%
\GlossariesWarningNoLine{No \string\printnoidxglossary\space
or \string\printnoidxglossaries ^^J
found. (Remove \string\makenoidxglossaries\space if you
don't want any glossaries.)^^JThis document will not have a glossary}%
}%
\let\warn@nomakeglossaries\relax
\let\makeglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
\let\@glo@assign@sortkey\@glo@assign@sortkey
\def\@glo@sorttype{\@glo@default@sorttype}%
}%
\renewcommand*\new@glossaryentry[2]{%
\PackageError{glossaries}{Glossary entries must be
defined in the preamble^^Jwhen you use
\string\makenoidxglossaries}%
{Either move your definitions to the preamble or use
\string\makeglossaries}%
}%
\renewcommand*{\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*\@gls@writefiles{%
  \forallglossaries{\@glo@type}{%
    \ifcsundef{glo@\@glo@type @filetok}%
    {%
      \def\gls@tmp{%
    }%
    }%
    {\edef\gls@tmp{\expandafter\the
      \csname glo@\@glo@type @filetok\endcsname}%
    }%
    \ifx\gls@tmp\@empty
      \ifx\@glo@type\glsdefaulttype
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries.^^JRemember to use package option 'nomain' if
you
          don't want to^^Juse the main glossary}%
      \else
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries}%
      \fi
    }%
  }%
}

```

```

        \fi
    \else
        \@glsopenfile{\glswrite}{\@glo@type}%
        \immediate\write\glswrite{%
            \expandafter\the
            \csname glo@\@glo@type @filetok\endcsname}%
        \immediate\closeout\glswrite
    \fi
}%
}
\newcommand*\@gls@glossary}[1]{%
    \@gls@glossary{#1}%
}
\newcommand*\@gls@glossary}[2]{%
    \if@gls@debug
        \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
    \fi
}
\newcommand*\@gls@renewglossary{%
    \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
    \let\@gls@renewglossary\@empty
}
\newcommand*\@gls@wrglossary}[2]{%
    \ifgls@savewrites
        \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
        \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
            \expandafter{\@gls@tmp~J}%
    \else
        \ifcsdef{glo@#1@file}%
            {%
                \expandafter\protected@write\csname glo@#1@file\endcsname{%
                    \gls@disablepagerefexpansion}{#2}%
            }%
            {%
                \ifignoredglossary{#1}{}%
                    {%
                        \GlossariesWarning{No file defined for glossary ‘#1’}%
                    }%
            }%
        \fi
    \endgroup\@esphack
}
\newcommand*\@do@wrglossary}[1]{%
    \glswriteentry{#1}{\@do@wrglossary{#1}}%
}
\newcommand*\@gls@writeentry}[2]{%
    \ifgls@indexonlyfirst
        \ifglsused{#1}{#2}%
    \else
        #2%
    \fi
}

```

```

\fi
}
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
\newcommand*{\gls@disablepagerefexpansion}{%
\@for\@gls@this:=\gls@protected@pagefmts\do
{%
\expandafter\let\@gls@this\relax
}%
}
\newcommand*{\gls@alphpage}{\@alph\c@page}
\newcommand*{\gls@Alphpage}{\@Alph\c@page}
\newcommand*{\gls@numberpage}{\number\c@page}
\newcommand*{\gls@arabicpage}{\@arabic\c@page}
\newcommand*{\gls@romanpage}{\romannumeral\c@page}
\newcommand*{\gls@Romanpage}{\@Roman\c@page}
\newcommand*{\glsaddprotectedpagefmt}[1]{%
\ea\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]{%
\ifglseclocations
\@do@esc@wrglossary{#1}%
\else
\@do@noesc@wrglossary{#1}%
\fi
}
\newcommand*{\@do@noesc@wrglossary}[1]{%
\expandafter\def\expandafter\@glslocref\expandafter{\theglentrycounter}%
\expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
\ifx\@glsHlocref\@glslocref
\def\@glo@counterprefix{}%
\else
\protected@edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
{\@glslocref}{\@glsHlocref}%
}%
\@do@gls@getcounterprefix

```



```

\fi
\edef\@gls@label{\glsdetoklabel{#1}}%
\@do@wrglossary
}
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodstrue
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
\begingroup
\let\gls@orgthe\the
\let\gls@orgnumber\number
\let\gls@orgarabic\@arabic
\let\gls@orgromannumeral\romannumeral
\let\gls@orgalph\@alph
\let\gls@orgAlph\@Alph
\let\gls@orgRoman\@Roman
\ifglswrallowprimitivemods
\def\gls@the##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
\def\the{\expandafter\gls@the}%
\def\gls@number##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
\def\number{\expandafter\gls@number}%
\fi
\def\@arabic##1{%
\ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\romannumeral##1{%
\ifx##1\c@page \gls@romanpage\else\gls@orgromannumeral##1\fi}%
\def\@Roman##1{%
\ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
\ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
\ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\@gls@locref{\thegl@entrycounter}%
\endgroup
\@gls@checkmkidxchars\@gls@locref
\expandafter\ifx\theHgl@entrycounter\thegl@entrycounter\relax
\def\@glo@counterprefix{}%
\else
\protected@edef\@gls@Hlocref{\theHgl@entrycounter}%
\@gls@checkmkidxchars\@gls@Hlocref
\edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
{\@gls@locref}{\@gls@Hlocref}%
}%
\@do@gls@getcounterprefix
\fi
\edef\@gls@label{\glsdetoklabel{#1}}%
\@do@wrglossary

```

```

}
\newcommand*{\@do@wrglossary}{%
  \ifglxindy
    \expandafter\@glo@check@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} \glsseeitem{#2}}
\newrobustcmd*{\glsseelist}[1]{%
  \let\@gls@dolast\relax
  \let\@gls@donext\relax
  \@for\@gls@thislabel:=#1\do{%
    \ifx\@xfor@nextelement\@nnil
      \@gls@dolast
    \else
      \@gls@donext
    \fi
    \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
    \let\@gls@dolast\glsseelastsep
    \let\@gls@donext\glsseesep
  }%
}
\newcommand*{\glsseelastsep}{\space\andname\space}
\newcommand*{\glsseesep}{, }

```

```

\newrobustcmd*{\glsseeitem}[1]{\gls hyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*{\glsseeitemformat}[1]{\glsentrytext{#1}}
\newcommand*{\gls@save@numberlist}[1]{%
  \ifgls@save@numberlist
    \toks@{#1}%
    \edef\@do@writeaux@info{%
      \noexpand\csgdef{glo@\gls@current@entry@label @numberlist}{\the\toks@}%
    }%
    \@onelevel@sanitize\@do@writeaux@info
    \protected@write\@auxout{}{\@do@writeaux@info}%
  \fi
}
\newcommand*{\warn@noprntglossary}{}%
\ifcsundef{printglossary}{}%
{
  \@gls@warn@gloss@defined
  \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}{%
  \forall@glossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}
\newcommand*{\printnoidxglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@noidx@glossary}%
}
\newcommand*{\printnoidxglossaries}{%
  \forall@glossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
\newcommand*{\@printgloss@setsort}{%
}
\newcommand*{\@gls@preglossaryhook}{%
}
\newcommand{\@printglossary}[2]{%

```

```

\def@glo@type{glsdefaulttype}%
\def@glossarytitle{\csname @glo@type @title\endcsname}%
\def@glossarytoctitle{glossarytitle}%
\let\org@glossarytitle@glossarytitle
\def@glossarystyle{%
  \ifx@glossary@default@style\relax
    \GlossariesWarning{No default glossary style provided \MessageBreak
      for the glossary '@glo@type'. \MessageBreak
      Using deprecated fallback. \MessageBreak
      To fix this set the style with \MessageBreak
      \string\setglossarystyle\space or use the \MessageBreak
      style key=value option}%
  \fi
}%
\def@gls@dotoc@title{glssettoctitle{@glo@type}}%
\let@org@glossaryentrynumbers@glossaryentrynumbers
\bgroup
  \@printgloss@setsort
  \setkeys{printgloss}{#1}%
  \@printgloss@checkexists{@glo@type}%
  {%
    \ifx@glossarytitle\org@glossarytitle
    \else
      \expandafter\let\csname @glo@type @title\endcsname
        \glossarytitle
    \fi
    \let\currentglossary@glo@type
    \let\org@glossaryentrynumbers@glossaryentrynumbers
    \let\glsnonextpages@glsnonextpages
    \let\glsnextpages@glsnextpages
    \let\nopostdesc@nopostdesc
    \gls@dotoc@title
    \@glossarystyle
    \let\gls@org@glossaryentryfield@glossentry
    \let\gls@org@glossarysubentryfield\subglossentry
    \renewcommand{\glossentry}[1]{%
      \xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
      \gls@org@glossaryentryfield{##1}%
    }%
    \renewcommand{\subglossentry}[2]{%
      \xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
      \gls@org@glossarysubentryfield{##1}{##2}%
    }%
    \@gls@preglossaryhook
    #2%
  }%
\egroup
\global\let@glossaryentrynumbers@org@glossaryentrynumbers
\global\let\warn@noprintglossary\relax
}

```

```

\newcommand{\@print@glossary}{%
  \makeatletter
  \@input@{\jobname.\csname @glo@type@\glo@type @in\endcsname}%
  \IfFileExists{\jobname.\csname @glo@type@\glo@type @in\endcsname}%
  {}%
  {\null}%
  \ifglxindy
    \ifcsundef{@xdy@\glo@type @language}%
    {%
      \edef\@do@auxoutstuff{%
        \noexpand\AtEndDocument{%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\providecommand\string\@xdylanguage[2]{}%
            \noexpand\immediate\noexpand\write\@auxout{%
              \string\@xdylanguage{\glo@type}{\@xdy@main@language}%
            }%
          }%
        }%
      }%
    }%
    {\edef\@do@auxoutstuff{%
      \noexpand\AtEndDocument{%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\providecommand\string\@xdylanguage[2]{}%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\@xdylanguage{\glo@type}{\csname @xdy@\glo@type
              @language\endcsname}%
          }%
        }%
      }%
    }%
  \do@auxoutstuff
  \edef\@do@auxoutstuff{%
    \noexpand\AtEndDocument{%
      \noexpand\immediate\noexpand\write\@auxout{%
        \string\providecommand\string\@gls@codepage[2]{}%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\@gls@codepage{\glo@type}{\gls@codepage}%
        }%
      }%
    }%
  \do@auxoutstuff
}
\fi
\renewcommand*{\@warn@nomakeglossaries}{%
  \GlossariesWarningNoLine{\string\makeglossaries\space
  hasn't been used,^^Jthe glossaries will not be updated}%
}%
}
\newcommand*{\@glo@sortentries}[2]{%
  \glosortentrieswarning
  \def\@glo@sortinglist{}%
  \def\@glo@sortinghandler{#1}%
  \edef\@glo@type{#2}%

```

```

\forlistcsloop{\@glo@do@sortentries}{@glsref@#2}%
\csdef{@glsref@#2}{}%
\@for\@this@label:=\@glo@sortinglist\do{%
  \xifinlistcs{\@this@label}{@glsref@#2}%
  {%
  {%
    \listcsxadd{@glsref@#2}{\@this@label}%
  }%
  \ifcsdef{@glo@sortingchildren@\@this@label}%
  {%
    \@glo@addchildren{#2}{\@this@label}%
  }%
  }%
}%
}
\newcommand*{\@glo@addchildren}[2]{%
  \bgroup
  \letcs{\@glo@childlist}{@glo@sortingchildren@#2}%
  \@for\@this@childlabel:=\@glo@childlist\do
  {%
    \xifinlistcs{\@this@childlabel}{@glsref@#1}%
    }%
    {%
    \listcsxadd{@glsref@#1}{\@this@childlabel}%
  }%
  \ifcsdef{@glo@sortingchildren@\@this@childlabel}%
  {%
    \@glo@addchildren{#1}{\@this@childlabel}%
  }%
  }%
  }%
  }%
  \egroup
}
\newcommand*{\@glo@do@sortentries}[1]{%
  \ifglshasparent{#1}%
  {%
  \edef\@glo@parent{\csuse{glo@\glsdetoklabel{#1}@parent}}%
  \ifcsundef{@glo@sortingchildren@\@glo@parent}%
  {%
    \csdef{@glo@sortingchildren@\@glo@parent}{}%
  }%
  }%
  \expandafter\@glo@sortedinsert
  \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
  \xifinlistcs{\@glo@parent}{@glsref@\@glo@type}%
  {%
  }%
  }%
  \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
}

```

```

    }%
  }%
  {%
    \@glo@sortedinsert{\@glo@sortinglist}{#1}%
  }%
}
\newcommand*{\@glo@sortedinsert}[2]{%
  \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
}%
\newcommand*{\@glo@sorthandler@word}[2]{%
  \letcs\@gls@sort@A{glo@glsetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@glsetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@letter}[2]{%
  \letcs\@gls@sort@A{glo@glsetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@glsetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@case}[2]{%
  \letcs\@gls@sort@A{glo@glsetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@glsetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@nocase}[2]{%
  \letcs\@gls@sort@A{glo@glsetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@glsetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlicompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sortmacro@word}[1]{%

```



```

\ifdefstring{\@glo@default@sorttype}{standard}%
{%
  \@glo@sortentries{\@glo@sorthandler@word}{#1}%
}%
{%
  \PackageError{glossaries}{Conflicting sort options:^^J
  \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
  \string\printnoidxglossary[sort=word]}{}}%
}%
}
\newcommand*\@glo@sortmacro@letter}[1]{%
\ifdefstring{\@glo@default@sorttype}{standard}%
{%
  \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
}%
{%
  \PackageError{glossaries}{Conflicting sort options:^^J
  \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
  \string\printnoidxglossary[sort=letter]}{}}%
}%
}
\newcommand*\@glo@sortmacro@standard}[1]{%
\ifdefstring{\@glo@default@sorttype}{standard}%
{%
  \ifcsdef{\@glo@sorthandler@\glsorder}%
  {%
    \@glo@sortentries{\csuse{\@glo@sorthandler@\glsorder}}{#1}%
  }%
  {%
    \PackageError{glossaries}{Unknown sort handler ‘\glsorder’}{}}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=standard]}{}}%
  }%
}
\newcommand*\@glo@sortmacro@case}[1]{%
\ifdefstring{\@glo@default@sorttype}{standard}%
{%
  \@glo@sortentries{\@glo@sorthandler@case}{#1}%
}%
{%
  \PackageError{glossaries}{Conflicting sort options:^^J
  \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
  \string\printnoidxglossary[sort=case]}{}}%
}%
}
\newcommand*\@glo@sortmacro@nocase}[1]{%

```

```

\ifdefstring{\@glo@default@sorttype}{standard}%
{%
  \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
}%
{%
  \PackageError{glossaries}{Conflicting sort options:^^J
  \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
  \string\printnoidxglossary[sort=nocase]}{}%
}%
}
\newcommand*{\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{%
  \forlssentries[#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@glstdetoklabel{#1}@loclist}%
\ifglshasparent{#1}%
{%
\gls@level=\csuse{glo@glstdetoklabel{#1}@level}\relax
\ifdefvoid{\@gls@loclist}
{%
\subglossentry{\gls@level}{#1}{}%
}%
}
}

```

```

    {%
      \subglossentry{\gls@level}{#1}%
      {%
        \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
      }%
    }%
  }%
  {%
    \letcs{\@gls@sort}{glo@\glsdetoklabel{#1}@sort}%
    \expandafter\glo@grabfirst\@gls@sort{}{}\@nil
    \ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
    {%
      \ifdefempty{\@gls@currentlettergroup}{}%
      {%
        \global\let\@glo@thislettergrp\@glo@thislettergrp
        \glsgroupskip
      }%
      \glsgroupheading{\@glo@thislettergrp}%
    }%
    \global\let\@gls@currentlettergroup\@glo@thislettergrp
    \ifdefvoid{\@gls@loclist}
    {%
      \glossentry{#1}{}%
    }%
    {%
      \glossentry{#1}%
      {%
        \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
      }%
    }%
  }%
}
}
\newcommand*{\glsnoidxloclist}[1]{%
  \def\@gls@noidxloclist@sep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxloclisthandler}{#1}%
}
\newcommand*{\glsnoidxloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%
    \@gls@noidxloclist@sep
    #1%
    \def\@gls@noidxloclist@sep{\delimN}%
    \def\@gls@noidxloclist@prev{#1}%
  }%
}
\newcommand*{\glsnoidxdisplayloclisthandler}[1]{%

```

```

\ifdefstring{\@gls@noidxloclist@prev}{#1}%
{%
}%
{%
  \@gls@noidxloclist@sep
  \@gls@noidxloclist@prev
  \def\@gls@noidxloclist@prev{#1}%
}%
}
\newcommand*\glsnoidxdisplayloc[4]{%
  \setentrycounter[#1]{#2}%
  \csuse{#3}{#4}%
}
\newcommand*\@gls@reference}[3]{%
  \glsdoifexistsorwarn{#2}%
  {%
    \ifcsundef{\@glsref@#1}{\csgdef{\@glsref@#1}{}}{}%
    \ifinlistcs{#2}{\@glsref@#1}%
    {}%
    {\listcsgadd{\@glsref@#1}{#2}}%
    \ifcsundef{glo@\glsdetoklabel{#2}@loclist}%
    {\csgdef{glo@\glsdetoklabel{#2}@loclist}{}}%
    {}%
    \listcsgadd{glo@\glsdetoklabel{#2}@loclist}{#3}%
  }%
}
\define@key{printgloss}{type}{\def\@glo@type{#1}}
\define@key{printgloss}{title}{%
  \def\glossarytitle{#1}%
  \let\gls@dotocitle\relax
}
\define@key{printgloss}{toctitle}{%
  \def\glossarytoctitle{#1}%
  \let\gls@dotocitle\relax
}
\define@key{printgloss}{style}{%
  \ifcsundef{\@glsstyle@#1}%
  {%
    \PackageError{glossaries}%
    {Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \def\@glossarystyle{\setglossentrycompatibility
      \csname @glsstyle@#1\endcsname}%
  }%
}
\define@choicekey{printgloss}{numberedsection}%
[\@gls@numberedsection@val\@gls@numberedsection@nr]%
{false,nolabel,autolabel,nameref}[nolabel]%
{}%

```

```

\ifcase\gls@numberedsection@nr\relax
  \renewcommand*{\@@glossarysecstar}{*}%
  \renewcommand*{\@@glossaryseclabel}{}%
\or
  \renewcommand*{\@@glossarysecstar}{}%
  \renewcommand*{\@@glossaryseclabel}{}%
\or
  \renewcommand*{\@@glossarysecstar}{}%
  \renewcommand*{\@@glossaryseclabel}{\label{\glsautoprefix\@glo@type}}%
\or
  \renewcommand*{\@@glossarysecstar}{*}%
  \renewcommand*{\@@glossaryseclabel}{%
    \protected@edef\@currentlabelname{\glossarytoctitle}%
    \label{\glsautoprefix\@glo@type}}%
\fi
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
  \csuse{glsnogroupskip#1}%
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
  \csuse{glsnopostdot#1}%
}
\newcommand*{\GlsEntryCounterLabelPrefix}{glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
  \csuse{glsentrycounter#1}%
  \@gls@define@glossaryentrycounter
}
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
  \csuse{glsesubentrycounter#1}%
  \@gls@define@glossarysubentrycounter
}
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
\ifglsnonumberlist
  \def\glossaryentrynumbers##1{%
\else
  \def\glossaryentrynumbers##1{##1}%
\fi}
\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
\newcommand*{\@glo@no@assign@sortkey}[1]{%
  \PackageError{glossaries}{‘sort’ key not permitted with
  \string\printglossary}%
  {The ‘sort’ key may only be used with \string\printnoidxglossary}%
}
\newcommand*{\@glo@assign@sortkey}[1]{%
  \def\@glo@sorttype{#1}%
}
\newcommand*{\@glsnonextpages}{%
  \gdef\glossaryentrynumbers##1{%
    \glsresetentrylist
  }%

```

```

}
\newcommand*{\@glsnextpages}{%
  \gdef\glossaryentrynumbers##1{%
    ##1\glsresetentrylist}}
\newcommand*{\glsresetentrylist}{%
  \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
\newcommand*{\glsnonextpages}{}
\newcommand*{\glsnextpages}{}
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
\newcommand*{\glsresetsubentrycounter}{%
  \ifglssubentrycounter
    \setcounter{glossarysubentry}{0}%
  \fi
}
\newcommand*{\glsresetentrycounter}{%
  \ifglentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*{\glsstepentry}[1]{%
  \ifglentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
\newcommand*{\glsstepsubentry}[1]{%
  \ifglssubentrycounter
    \edef\currentglssubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
  \fi
}
\newcommand*{\glsrefentry}[1]{%
  \ifglentrycounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else
    \ifglssubentrycounter
      \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
      \gls{#1}%
    \fi
  \fi
}
\newcommand*{\glsentrycounterlabel}{%
  \ifglentrycounter
    \theglossaryentry.\space
  \fi
}
\newcommand*{\glssubentrycounterlabel}{%

```

```

\ifglssubentrycounter
  \theglossarysubentry)\space
\fi
}
\newcommand*\glssentryitem}[1]{%
  \ifglssentrycounter
    \glssstepentry{#1}\glssentrycounterlabel
  \else
    \glssresetsubentrycounter
  \fi
}
\newcommand*\glsssubentryitem}[1]{%
  \ifglsssubentrycounter
    \glssstepsubentry{#1}\glsssubentrycounterlabel
  \fi
}
\ifcsundef{theglossary}%
  {%
    \newenvironment{theglossary}{}{}%
  }%
  {%
    \@gls@warnontheGLOSSDEFINED
    \renewenvironment{theglossary}{}{}%
  }
\newcommand*\glossaryheader{}
\newcommand*\glstarget}[2]{\@glstarget{\glo@linkprefix#1}{#2}}
\providecommand*\compatibleglossentry}[2]{%
  \toks@{#2}%
  \protected@edef\@do@glssentry{\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@glssentry
}
\newcommand*\glossentryname}[1]{%
  \glssdoifexistsorwarn{#1}%
  {%
    \letcs{\glo@name}{glo@\glssdetoklabel{#1}@name}%
    \expandafter\glsnamefont\expandafter{\glo@name}%
  }%
}
\newcommand*\Glossentryname}[1]{%
  \glssdoifexistsorwarn{#1}%
  {%
    \glsnamefont{\Glsentryname{#1}}%
  }%
}
}

```



```

\newcommand*\glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrydesc{#1}%
  }%
}
\newcommand*\Glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrydesc{#1}%
  }%
}
\newcommand*\glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrysymbol{#1}%
  }%
}
\newcommand*\Glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrysymbol{#1}%
  }%
}
\providecommand*\compatiblesubglossentry}[3]{%
  \toks@{#3}%
  \protected@edef\@do@subglossentry{\noexpand\glossarysubentryfield{\number#1}%
  {#2}%
  {\noexpand\glsnamefont
    {\expandafter\expandonce\csname glo@#2@name\endcsname}}%
  {\expandafter\expandonce\csname glo@#2@desc\endcsname}}%
  {\expandafter\expandonce\csname glo@#2@symbol\endcsname}}%
  {\the\toks@}%
  }%
  \@do@subglossentry
}
\newcommand*\setglossentrycompatibility){%
  \let\glossentry\compatibleglossentry
  \let\subglossentry\compatiblesubglossentry
}
\setglossentrycompatibility
\newcommand*\glossaryentryfield}[5]{%
  \GlossariesWarning
  {Deprecated use of \string\glossaryentryfield.^~J
  I recommend you change to \string\glossentry.^~J
  If you've just upgraded, try removing your gls auxiliary
  files^~J and recompile}%
  \noindent\textbf{\glstarget{#1}{#2}} #4 #3. #5\par}
\newcommand*\glossarysubentryfield}[6]{%
  \GlossariesWarning

```

```

{Deprecated use of \string\glossarysubentryfield.^^J
 I recommend you change to \string\subglossentry.^^J
 If you've just upgraded, try removing your gls auxiliary
 files^^J and recompile}%
\glstarget{#2}{\strut}#4. #6\par}
\newcommand*{\glsgroupskip}{}
\newcommand*{\glsgroupheading}[1]{}
\newcommand*{\glsgetgrouptitle}[1]{%
 \@gls@getgrouptitle{#1}{\@gls@grptitle}%
 \@gls@grptitle
}
\newcommand*{\@gls@getgrouptitle}[2]{%
 \dtl@ifsingle{#1}%
 {%
 \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
 }%
 {%
 \ifboolexpr{test{\ifstrequal{#1}{glsymbols}}
 or test{\ifstrequal{#1}{glsnumbers}}}%
 {%
 \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
 }%
 {%
 \def#2{#1}%
 }%
 }%
}
}
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%
 \DTLifint{#1}%
 {\edef#2{\char#1\relax}}%
 {%
 \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
 }%
}
}
\newcommand*{\glsgetgrouplabel}[1]{%
 \ifthenelse{\equal{#1}{\glsymbolsgroupname}}{\glsymbols}{%
 \ifthenelse{\equal{#1}{\glsnumbersgroupname}}{\glsnumbers}{#1}}%
}
\newcommand*{\setentrycounter}[2][{}]{%
 \def\@glo@counterprefix{#1}%
 \ifx\@glo@counterprefix\@empty
 \def\@glo@counterprefix{.}%
 \else
 \def\@glo@counterprefix{.#1}%
 \fi
 \def\glsentrycounter{#2}%
}
}
\newcommand*{\setglossarystyle}[1]{%
 \ifcsundef{@glsstyle@#1}%
 {%
 \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
}
}

```

```

}%
{%
  \csname @glsstyle@#1\endcsname
}%
\ifx\@glossary@default@style\relax
  \protected@edef\@glossary@default@style{#1}%
\fi
}
\newcommand*\@glossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \GlossariesWarning
    {Deprecated command \string\glossarystyle.^^J
     I recommend you switch to \string\setglossarystyle\space unless
     you want to maintain backward compatibility}%
    \setglossentrycompatibility
    \csname @glsstyle@#1\endcsname
    \ifcsdef{@glscompstyle@#1}%
    {\setglossentrycompatibility\cuse{@glscompstyle@#1}}%
    {}%
  }%
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\newcommand*\@newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
\newcommand*\@renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
\newcommand*\@glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\glsnumber#1{#1}%

```

```

}%
{%
  \def\glsnumber#1{\@glsnumber#1\nohyperpage{}}\@nil}
}
\def\@glsnumber#1\nohyperpage#2#3\@nil{%
  \ifx\#1\%
  \else
    \@delimR#1\delimR\delimR\%
  \fi
  \ifx\#2\%
  \else
    #2%
  \fi
  \ifx\#3\%
  \else
    \@glsnumber#3\@nil
  \fi
}
\def\@delimR#1\delimR #2\delimR #3\%
\ifx\#2\%
  \@delimN{#1}%
\else
  \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
\fi}
\def\@delimN#1{\@delimN#1\delimN \delimN\}
\def\@delimN#1\delimN #2\delimN#3\%
\ifx\#3\%
  \@gls@numberlink{#1}%
\else
  \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
\fi
}
\def\@gls@numberlink#1{%
\begingroup
\toks@={}%
\@gls@removespaces#1 \@nil
\endgroup}
\def\@gls@removespaces#1 #2\@nil{%
\toks@=\expandafter{\the\toks@#1}%
\ifx\#2\%
  \edef\x{\the\toks@}%
  \ifx\x\empty
  \else
    \hyperlink{\glsentrycounter\@gls@counterprefix\the\toks@}%
      {\the\toks@}%
  \fi
\else
  \@gls@ReturnAfterFi{%
    \@gls@removespaces#2\@nil
  }%
}

```

```

\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
\newcommand*\hyperrm[1]{\textrm{\glsnumber{#1}}}
\newcommand*\hypersf[1]{\textsf{\glsnumber{#1}}}
\newcommand*\hypertt[1]{\texttt{\glsnumber{#1}}}
\newcommand*\hyperbf[1]{\textbf{\glsnumber{#1}}}
\newcommand*\hypermd[1]{\textmd{\glsnumber{#1}}}
\newcommand*\hyperit[1]{\textit{\glsnumber{#1}}}
\newcommand*\hypersl[1]{\textsl{\glsnumber{#1}}}
\newcommand*\hyperup[1]{\textup{\glsnumber{#1}}}
\newcommand*\hypersc[1]{\textsc{\glsnumber{#1}}}
\newcommand*\hyperemph[1]{\emph{\glsnumber{#1}}}
\newcommand{\oldacronym}[4][\gls@label]{%
  \def\gls@label{#2}%
  \newacronym[4]{#1}{#2}{#3}%
  \ifcsundef{xspace}%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}%
    }%
  }%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
        \noexpand\gls{#1}\noexpand\xspace}%
      }%
  }%
}
\newcommand{\newacronym}[4][{}]{
\newcommand*\acrpluralsuffix{\glsacrpluralsuffix}
\newrobustcmd*\glsstextup[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
\newcommand*\glsshortkey{short}
\newcommand*\glsshortpluralkey{shortplural}
\newcommand*\glslongkey{long}
\newcommand*\glslongpluralkey{longplural}
\newrobustcmd*\acrfull{\@gls@hyp@opt\ns@acrfull}
\newcommand*\ns@acrfull[2][{}]{%
  \new@ifnextchar[{\@acrfull{#1}{#2}}%
  {\@acrfull{#1}{#2} []}%
}
\def\@acrfull#1#2[#3]{%
  \acrfullfmt{#1}{#2}{#3}%
}
\newcommand*\acrfullfmt[3]{%
  \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newcommand{\acrlinkfullformat}[5]{%
  \acrfullformat{#1}{#3}{#4}[#5]{#2}{#3}{#4} []}%
}

```

```

\newcommand{\acrfullformat}[2]{#1\glsspace(#2)}
\newrobustcmd{\glsspace}{\space}
\newrobustcmd*{\Acrfull}{\@gls@hyp@opt\ns@Acrfull}
\newcommand*\ns@Acrfull[2][]{%
  \new@ifnextchar[{\@Acrfull{#1}{#2}}{%
    {\@Acrfull{#1}{#2} []}%
  }
}
\def\@Acrfull#1#2[#3]{%
  \Acrfullfmt{#1}{#2}{#3}%
}
\newcommand*{\Acrfullfmt}[3]{%
  \acrlinkfullformat{\@Acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfull}{\@gls@hyp@opt\ns@ACRfull}
\newcommand*\ns@ACRfull[2][]{%
  \new@ifnextchar[{\@ACRfull{#1}{#2}}{%
    {\@ACRfull{#1}{#2} []}%
  }
}
\def\@ACRfull#1#2[#3]{%
  \ACRfullfmt{#1}{#2}{#3}%
}
\newcommand*{\ACRfullfmt}[3]{%
  \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfullpl}{\@gls@hyp@opt\ns@acrfullpl}
\newcommand*\ns@acrfullpl[2][]{%
  \new@ifnextchar[{\@acrfullpl{#1}{#2}}{%
    {\@acrfullpl{#1}{#2} []}%
  }
}
\def\@acrfullpl#1#2[#3]{%
  \acrfullplfmt{#1}{#2}{#3}%
}
\newcommand*{\acrfullplfmt}[3]{%
  \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*{\Acrfullpl}{\@gls@hyp@opt\ns@Acrfullpl}
\newcommand*\ns@Acrfullpl[2][]{%
  \new@ifnextchar[{\@Acrfullpl{#1}{#2}}{%
    {\@Acrfullpl{#1}{#2} []}%
  }
}
\def\@Acrfullpl#1#2[#3]{%
  \Acrfullplfmt{#1}{#2}{#3}%
}
\newcommand*{\Acrfullplfmt}[3]{%
  \acrlinkfullformat{\@Acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfullpl}{\@gls@hyp@opt\ns@ACRfullpl}
\newcommand*\ns@ACRfullpl[2][]{%
  \new@ifnextchar[{\@ACRfullpl{#1}{#2}}{%
    {\@ACRfullpl{#1}{#2} []}%
  }
}

```

```

}
\def\ACRfullpl#1#2[#3]{%
  \ACRfullplfmt{#1}{#2}{#3}%
}
\newcommand*\ACRfullplfmt}[3]{%
  \acrlinkfullformat{\ACRlongpl}{\ACRshortpl}{#1}{#2}{#3}%
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand*\acrnameformat}[2]{\acronymfont{#1}}
\newtoks\glskeylisttok
\newtoks\glslabeltok
\newtoks\glsshorttok
\newtoks\gslongtok
\newcommand*\newacronymhook{}
\newcommand*\SetGenericNewAcronym}{%
  \let\@Gls@entryname\@Gls@acrenryname
  \renewcommand{\newacronym}[4][]{%
    \ifdefempty{\@glsacronymlists}%
      {%
        \def\@glo@type{\acronymtype}%
        \setkeys{glossentry}{##1}%
        \DeclareAcronymList{\@glo@type}%
      }%
    }%
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \gslongtok{##4}%
  \newacronymhook
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\expandonce{\acronymentry{##2}}},%
      sort={\acronymsort{\the\glsshorttok}{\the\gslongtok}},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\gslongtok},%
      longplural={\the\gslongtok\noexpand\acrpluralsuffix},%
      \GenericAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}%
\renewcommand*\acrfullfmt}[3]{%
  \glslink{##1}{##2}{\genacrfullformat{##2}{##3}}%
\renewcommand*\Acrfullfmt}[3]{%

```

```

\glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
\renewcommand*\ACRfullfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}%
\renewcommand*\acrfullplfmt}[3]{%
\glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}%
\renewcommand*\glsentryfull}[1]{\genacrfullformat{##1}{}}}%
\renewcommand*\Glsentryfull}[1]{\Genacrfullformat{##1}{}}}%
\renewcommand*\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}}%
\renewcommand*\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}}%
}
\newcommand*\GenericAcronymFields){description={\the\glslongtok}}
\newcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}
\newcommand*\acronymsort}[2]{##1}
\newcommand*\setacronymstyle}[1]{%
\ifcsundef{@glsacr@dispstyle@##1}
{%
\PackageError{glossaries}{Undefined acronym style ‘##1’}{%
}%
}%
\ifdefempty{\@glsacronymlists}%
{%
\DeclareAcronymList{\acronymtype}%
}%
}%
\SetGenericNewAcronym
\GlsUseAcrStyleDefs{##1}%
\@for\@gls@type:=\@glsacronymlists\do{%
\defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{##1}}%
}%
}%
}
\newcommand*\newacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@##1}%
{%
\PackageError{glossaries}{Acronym style ‘##1’ already exists}{%
}%
}%
\csdef{@glsacr@dispstyle@##1}{##2}%
\csdef{@glsacr@styledefs@##1}{##3}%
}%
}
\newcommand*\renewacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@##1}%
{%

```



```

\csdef{@glsacr@dispstyle@#1}{#2}%
\csdef{@glsacr@styledefs@#1}{#3}%
}%
{%
\PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
}%
}
\newcommand*{\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*{\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
{%
\ifglshaslong{\glslabel}{\glsacrfmt}{\glsacrfmt}}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
\ifglshaslong{\glslabel}{\glsacrfmt}{\glsacrfmt}}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}

```

```

}%
\renewcommand*\genplacrfullformat}[2]{%
  \glentrylongpl{##1}##2\glsacspace{##1}%
  (\protect\firstacronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
  \Glentrylongpl{##1}##2\glsacspace{##1}%
  (\protect\firstacronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newcommand*\glsacspace}[1]{%
  \settowidth{\dimen@}{(\protect\firstacronymfont{\glentryshort{##1}})}%
  \ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
  \ifglshaslong{\glslabel}{\glsacspace}{\glsacspace}%
}%
{%
  \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
  \renewcommand*\genacrfullformat}[2]{%
    \protect\firstacronymfont{\glentryshort{##1}}##2\space
    (\glentrylong{##1})%
  }%
  \renewcommand*\Genacrfullformat}[2]{%
    \protect\firstacronymfont{\Glentryshort{##1}}##2\space
    (\glentrylong{##1})%
  }%
  \renewcommand*\genplacrfullformat}[2]{%
    \protect\firstacronymfont{\glentryshortpl{##1}}##2\space
    (\glentrylongpl{##1})%
  }%
  \renewcommand*\Genplacrfullformat}[2]{%
    \protect\firstacronymfont{\Glentryshortpl{##1}}##2\space
    (\glentrylongpl{##1})%
  }%
  \renewcommand*\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
  \renewcommand*\acronymsort}[2]{##1}%
  \renewcommand*\acronymfont}[1]{##1}%
  \renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
  \renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}

```

```

}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{long-sm-short}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{sc-short-long}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{sm-short-long}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sp-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sp-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sp-short}%
}

```

```

\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glentrylong{##1}\glsacspace{##1}(\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sc-short-desc}%
{
  \GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{
  \GlsUseAcrStyleDefs{long-sc-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{long-sm-short-desc}%
{
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{short-long-desc}%
{
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sc-short-long-desc}%
{
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sm-short-long-desc}%

```

```

{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \Glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \mfirstucMakeUppercase
            {\glsentrylongpl{\glslabel}\glsinsert}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \Glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \mfirstucMakeUppercase
          {\glsentrylong{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glsgenentryfmt
  }%
  {\glscustomtext\glsinsert}%
}%

```

```

{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\glsentrylong{##2}##3\space}
  (\acronymfont{\glsentryshort{##2}})}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Glsentrylong{##2}##3\space}
  (\acronymfont{\glsentryshort{##2}})}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\glsentrylong{##2}##3\space}
    (\acronymfont{\glsentryshort{##2}})}%
  }%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\glsentrylongpl{##2}##3\space}
  (\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Glsentrylongpl{##2}##3\space}
  (\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\glsentrylongpl{##2}##3\space}
    (\acronymfont{\glsentryshortpl{##2}})}%
  }%
\renewcommand*{\glsentryfull}[1]{%
  \glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
  \glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
  \GlsUseAcrEntryDispStyle{dua}%
}%
{%
  \GlsUseAcrStyleDefs{dua}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentrylong{##1}}}%
  \renewcommand*{\acronymsort}[2]{##2}%
}%
\newacronymstyle{footnote}%

```

```

{%
\ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshort{##1}}##2%
\protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\firstacronymfont{\Glsentryshort{##1}}##2%
\protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshortpl{##1}}##2%
\protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
\protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
(\glsentrylong{##2})}%
\renewcommand*{\Acrfullfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
(\glsentrylong{##2})}%
\renewcommand*{\ACRfullfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\acronymfont{\glsentryshort{##2}}##3\space
(\glsentrylong{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
(\glsentrylongpl{##2})}%
\renewcommand*{\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
(\glsentrylongpl{##2})}%
\renewcommand*{\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{##2}}##3\space
(\glsentrylongpl{##2})}}}%
\renewcommand*{\glsentryfull}[1]{%
\acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
\acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%

```

```

\renewcommand*\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]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newcommand*\DefineAcronymSynonyms{%
  \let\acs\acrshort
  \let\Acs\Acrshort
  \let\acsp\acrshortpl
  \let\Acp\Acrshortpl
  \let\acl\acrlong
  \let\Acl\Acrlong
  \let\aclp\acrlongpl
  \let\Aclp\Acrlongpl
  \let\acf\acrfull
  \let\Acf\Acrfull
  \let\acfp\acrfullpl
  \let\Acfp\Acrfullpl
  \let\ac\gls
  \let\Ac\Gls
  \let\acp\glspl
  \let\Acp\Glspl
}
\ifglsacrshortcuts
  \DefineAcronymSynonyms
\fi
\newcommand*\SetDefaultAcronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsentryfmt}%
}
\newcommand*\DefaultNewAcronymDef{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      sort={\the\glsshorttok},%
      text={\the\glsshorttok},%
      first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
      plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      firstplural={\acrfullformat{\noexpand\expandonce\noexpand\@glo@longpl}%
        {\noexpand\expandonce\noexpand\@glo@shortpl}},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
    }
  }
}

```

```

        \the\glskeylisttok
    }%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
    \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
    \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
    \@@gls@expand@field{##1}{descplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*\@SetDefaultAcronymStyle{%
    \@for\@gls@type:=\@gls@acronymlists\do{%
        \@SetDefaultAcronymDisplayStyle{\@gls@type}%
    }%
\renewcommand*\@newacronym[4][]{%
    \ifx\@gls@acronymlists\@empty
        \def\@glo@type{\acronymtype}%
        \setkeys{glossentry}{##1}%
        \DeclareAcronymList{\@glo@type}%
        \@SetDefaultAcronymDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \DefaultNewAcronymDef
}%
\renewcommand*\@acrpluralsuffix{\@gls@acrpluralsuffix}%
}
\newcommand*\@acrfootnote[3]{\@acrlinkfootnote{#1}{#2}{#3}}
\newcommand*\@acrlinkfootnote[3]{%
    \footnote{\@glslink[##1]{##2}{##3}}%
}
\newcommand*\@acrnofootnote[3]{%
    \footnote{##3}%
}
\newcommand*\@SetDescriptionFootnoteAcronymDisplayStyle[1]{%
    \def\glsentryfmt[##1]{%
        \ifdefempty\gls@customtext

```

```

{%
  \ifglused{\glslabel}%
  {%
    \acronymfont{\glsgetentryfmt}%
  }%
  {%
    \firstacronymfont{\glsgetentryfmt}%
    \ifglshassymbol{\glslabel}%
    {%
      \expandafter\protect\expandafter\acrfootnote\expandafter
      {\@gls@link@opts}{\@gls@link@label}%
      {%
        \glsifplural
        {\glsentrysymbolplural{\glslabel}}%
        {\glsentrysymbol{\glslabel}}%
      }%
    }%
  }%
}
}
\newcommand*{\DescriptionFootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\glsshorttok}},%
    sort={\the\glsshorttok},%
    first={\the\glsshorttok},%
    firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
    text={\the\glsshorttok},%
    plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
    short={\the\glsshorttok},%
    shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
    long={\the\glslongtok},%
    longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    symbol={\the\glslongtok},%
    symbolplural={\noexpand\expandonce\noexpand\@glo@longpl},%
    \the\glskeylisttok
  }%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}

```

```

}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*\SetDescriptionFootnoteAcronymStyle{%
  \renewcommand{\newacronym}[4][{}]{%
    \ifx\@glsacronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetDescriptionFootnoteAcronymDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \DescriptionFootnoteNewAcronymDef
  }%
  \@for\@gls@type:=\@glsacronymlists\do{%
    \SetDescriptionFootnoteAcronymDisplayStyle{\@gls@type}%
  }%
  \ifglsacrsmallcaps
    \renewcommand*\acronymfont[1]{\textsc{##1}}%
    \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
  \else
    \ifglsacrsmaller
      \renewcommand*\acronymfont[1]{\textsmaller{##1}}%
    \fi
  \fi
  \ifglsacrdua
    \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
      can’t both be set}{}%
  \fi
}%
\newcommand*\SetDescriptionDUAACronymDisplayStyle[1]{%
  \defglsentryfmt[##1]{\glsgenentryfmt}%
}
\newcommand*\DescriptionDUANewAcronymDef{%
  \edef\do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glslongtok},%
      sort={\the\glslongtok},

```

```

text={\the\glslongtok},%
first={\the\glslongtok},%
plural={\noexpand\expandonce\noexpand\@glo@longpl},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
symbol={\the\glsshorttok},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionDUAAcronymStyle}{%
  \ifglsacrsmallcaps
    \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
      can’t both be set}{}%
  \else
    \ifglsacrsmaller
      \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
        can’t both be set}{}%
    \fi
  \fi
\renewcommand{\newacronym}[4][[]]{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetDescriptionDUAAcronymDisplayStyle{\@glo@type}%
  \fi
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%

```

```

\glslongtok{##4}%
\newacronymhook
\DescriptionDUANewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetDescriptionDUAAcronymDisplayStyle{\@gls@type}%
}%
}%
\newcommand*\SetDescriptionAcronymDisplayStyle[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\ifglsused{\glslabel}%
{%
\let\gls@org@insert\glsinsert
\let\glsinsert\@empty
\acronymfont{\glsentryfmt}\gls@org@insert
}%
{%
\glsentryfmt
\ifglsymbol{\glslabel}%
{%
\glsifplural
{%
\def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%
}%
{%
\def\@glo@symbol{\glsentrysymbol{\glslabel}}%
}%
\space(\protect\firstacronymfont
{\gls@symbol}
{\@glo@symbol}
{\mfirstucMakeUppercase{\@glo@symbol}})}%
}%
{}}%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*\DescriptionNewAcronymDef){%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand
\acrnameformat{\the\glsshorttok}{\the\glslongtok}},%
sort={\the\glsshorttok},%
first={\the\glslongtok},%

```

```

firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
symbol={\noexpand\@glo@text},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionAcronymStyle}{%
\renewcommand{\newacronym}[4][]{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetDescriptionAcronymDisplayStyle{\@glo@type}%
  \fi
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \glslongtok{##4}%
  \newacronymhook
  \DescriptionNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
  \SetDescriptionAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
\renewcommand{\acronymfont}[1]{\textsc{##1}}
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
\else

```

```

\ifglsmaller
\renewcommand*\acronymfont}[1]{\textsmaller{##1}}%
\fi
\fi
}%
\newcommand*\SetFootnoteAcronymDisplayStyle}[1]{%
\defglentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\let\gls@org@insert\glsinsert
\let\glsinsert\@empty
\ifglsused{\glslabel}%
{%
\acronymfont{\glsentryfmt}\gls@org@insert
}%
{%
\firstacronymfont{\glsentryfmt}\gls@org@insert
\ifglshaslong{\glslabel}%
{%
\expandafter\protect\expandafter\acrfootnote\expandafter
{\@gls@link@opts}{\@gls@link@label}%
}%
\glsifplural
{\glsentrylongpl{\glslabel}}%
{\glsentrylong{\glslabel}}%
}%
}%
}%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*\FootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
first={\the\glsshorttok},%
firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\the\glslongtok},%
descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%

```



```

        \the\glskeylisttok
    }%
}%
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@@gls@expand@field{##1}{descplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@descplural\@org@gls@assign@descplural
}
\newcommand*\SetFootnoteAcronymStyle{%
  \renewcommand*\newacronym[4][ ]{%
    \ifx\@glsacronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetFootnoteAcronymDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \FootnoteNewAcronymDef
  }%
  \for\@gls@type:=\@glsacronymlists\do{%
    \SetFootnoteAcronymDisplayStyle{\@gls@type}%
  }%
  \ifglsacrsmallcaps
    \renewcommand*\acronymfont[1]{\textsc{##1}}%
    \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
  \else
    \ifglsacrsmaller
      \renewcommand*\acronymfont[1]{\textsmaller{##1}}%
    \fi
  \fi
  \ifglsacrdua
    \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
    can’t both be set}{%
  \fi

```

```

}%
\DeclareRobustCommand*{\glsdoparenifnotempty}[2]{%
  \protected@edef\gls@tmp{#1}%
  \ifdefempty\gls@tmp
  {}%
  {%
    \ifx\gls@tmp\@gls@default@value
    \else
    \space (#2{#1})%
    \fi
  }%
}
}
\newcommand*{\SetSmallAcronymDisplayStyle}[1]{%
  \def\glsentryfmt[#1]{%
    \ifdefempty\gls@customtext
    {%
      \let\gls@org@insert\glsinsert
      \let\glsinsert\@empty
      \ifglsused{\glslabel}%
      {%
        \acronymfont{\gls@genentryfmt}\gls@org@insert
      }%
      {%
        \gls@genentryfmt
        \ifgls@hassymbol{\glslabel}%
        {%
          \gls@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\glsshorttok}},%
sort={\the\glsshorttok},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
first={\the\glslongtok},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\noexpand\@glo@first},%
descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
symbol={\the\glsshorttok},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@@gls@expand@field{##1}{descplural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@descplural\@org@gls@assign@descplural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetSmallAcronymStyle}{%
\renewcommand{\newacronym}[4][{}]{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetSmallAcronymDisplayStyle{\@glo@type}%
  \fi
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
}

```

```

\glslongtok{##4}%
\newacronymhook
\SmallNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetSmallAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
\renewcommand*\acronymfont}[1]{\textsc{##1}}
\renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
\else
\renewcommand*\acronymfont}[1]{\textsmaller{##1}}
\fi
\ifglsacrdua
\ifglsacrsmallcaps
\PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
can’t both be set}{}%
\else
\PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
can’t both be set}{}%
\fi
\fi
}%
\newcommand*\SetDUADisplayStyle}[1]{%
\defglsentryfmt[#1]{\glsentryfmt}%
}
\newcommand*\DUANewAcronymDef{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\the\glsshorttok},%
text={\the\glslongtok},%
first={\the\glslongtok},%
plural={\noexpand\expandonce\noexpand\@glo@longpl},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\the\glslongtok},%
descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
symbol={\the\glsshorttok},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural

```

```

\let\org@gl@assign@descplural\gl@assign@descplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@symbolplural##1##2{%
  \@gl@expand@field{##1}{symbolplural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@gl@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\org@gl@assign@firstpl
\let\gl@assign@plural\org@gl@assign@plural
\let\gl@assign@symbolplural\org@gl@assign@symbolplural
\let\gl@assign@descplural\org@gl@assign@descplural
}
\newcommand*{\SetDUASStyle}{%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@gl@sacronymlists\@empty
      \def\@gl@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@gl@type}%
      \SetDUADisplayStyle{\@gl@type}%
    \fi
    \gl@keylisttok{##1}%
    \gl@labeltok{##2}%
    \gl@shorttok{##3}%
    \gl@longtok{##4}%
    \newacronymhook
    \DUANewAcronymDef
  }%
  \@for\@gl@type:=\@gl@sacronymlists\do{%
    \SetDUADisplayStyle{\@gl@type}%
  }%
}
\newcommand*{\SetAcronymStyle}{%
  \SetDefaultAcronymStyle
  \ifgl@acrdescription
    \ifgl@sacrfootnote
      \SetDescriptionFootnoteAcronymStyle
    \else
      \ifgl@sacrdua
        \SetDescriptionDUAAcronymStyle
      \else
        \SetDescriptionAcronymStyle
      \fi
    \fi
  \fi
}

```

```

\else
  \ifglacrfootnote
    \SetFootnoteAcronymStyle
  \else
    \ifthenelse{\boolean{glacrsmallcaps}\OR
      \boolean{glacrsmaller}}{%
      {%
        \SetSmallAcronymStyle
      }%
    }%
    \ifglacrdua
      \SetDUASStyle
    \fi
  }%
\fi
}
\SetAcronymStyle
\newcommand*{\SetCustomDisplayStyle}[1]{%
  \defglentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*{\CustomAcronymFields}{%
  name={\the\glsshorttok},%
  description={\the\glslongtok},%
  first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
  firstplural={\acrfullformat
    {\noexpand\glsentrylongpl{\the\glslabeltok}}%
    {\noexpand\glsentryshortpl{\the\glslabeltok}}},%
  text={\the\glsshorttok},%
  plural={\the\glsshorttok\noexpand\acrpluralsuffix}%
}
\newcommand*{\CustomNewAcronymDef}{%
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=acronymtype,%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      user1={\the\glsshorttok},%
      user2={\the\glsshorttok\noexpand\acrpluralsuffix},%
      user3={\the\glslongtok},%
      user4={\the\glslongtok\noexpand\acrpluralsuffix},%
      \CustomAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}

```

```

\newcommand*\SetCustomStyle{%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@glsacronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetCustomDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \CustomNewAcronymDef
  }%
  \@for\@gls@type:=\@glsacronymlists\do{%
    \SetCustomDisplayStyle{\@gls@type}%
  }%
}
\RequirePackage{glossary-hypernav}[=v4.46]
\@gls@loadlist
\@gls@loadlong
\@gls@loadsuper
\@gls@loadtree
\ifx\@glossary@default@style\relax
\else
  \setglossarystyle{\@glossary@default@style}
\fi
\newcommand*\showgloparent}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
}
\newcommand*\showglolevel}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
}
\newcommand*\showglotext}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
}
\newcommand*\showgloplural}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@plural\endcsname
}
\newcommand*\showglofirst}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@first\endcsname
}
\newcommand*\showglofirstpl}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpl\endcsname
}
\newcommand*\showglotype}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@type\endcsname
}
\newcommand*\showglocounter}[1]{%

```

```

\expandafter\show\csname glo@\glsdetoklabel{#1}@counter\endcsname
}
\newcommand*{\showgluseri}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@useri\endcsname
}
\newcommand*{\showgluserii}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@userii\endcsname
}
\newcommand*{\showgluseriii}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@useriii\endcsname
}
\newcommand*{\showgluseriv}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@useriv\endcsname
}
\newcommand*{\showgluserv}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@userv\endcsname
}
\newcommand*{\showgluservi}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@uservi\endcsname
}
\newcommand*{\showgloname}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@name\endcsname
}
\newcommand*{\showglodesc}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@desc\endcsname
}
\newcommand*{\showglodescplural}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@descplural\endcsname
}
\newcommand*{\showglosort}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@sort\endcsname
}
\newcommand*{\showglosymbol}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@symbol\endcsname
}
\newcommand*{\showglosymbolplural}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@symbolplural\endcsname
}
\newcommand*{\showgloshort}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@short\endcsname
}
\newcommand*{\showglolong}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname
}
\newcommand*{\showgloindex}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname
}
\newcommand*{\showgloflag}[1]{%
\expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname
}
}

```



```

\newcommand*\showgloloclist}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@loclist\endcsname
}
\newcommand*\showglofield}[2]{%
  \csshow{glo@glstdetoklabel{#1}@#2}%
}
\newcommand*\showacronymlists}{%
  \show@glssacronymlists
}
\newcommand*\showglossaries}{%
  \show@glo@types
}
\newcommand*\showglossaryin}[1]{%
  \expandafter\show\csname @glotype@#1@in\endcsname
}
\newcommand*\showglossaryout}[1]{%
  \expandafter\show\csname @glotype@#1@out\endcsname
}
\newcommand*\showglossarytitle}[1]{%
  \expandafter\show\csname @glotype@#1@title\endcsname
}
\newcommand*\showglossarycounter}[1]{%
  \expandafter\show\csname @glotype@#1@counter\endcsname
}
\newcommand*\showglossaryentries}[1]{%
  \expandafter\show\csname glo@list@#1\endcsname
}
\csname ifglsc-compatible-2.07\endcsname
  \RequirePackage{glossaries-compatible-207}[v4.46]
\fi

```

8.2 Rollback v4.46 (glossaries-accsupp-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-accsupp}[2020/03/19 v4.46 (NLCT)
  Experimental glossaries accessibility]
\DeclareOption*\PassOptionsToPackage{\CurrentOption}{glossaries}}
\ProcessOptions
\@ifpackageloaded{glossaries-extra}
{%
  \ifx@glsxtr@doaccsupp\empty
  \GlossariesWarning{The 'glossaries-accsupp'
    package has been loaded\MessageBreak
    after the 'glossaries-extra' package. This\MessageBreak
    can cause a failure to integrate both packages. \MessageBreak
    Either use the 'accsupp' option when you load\MessageBreak
    'glossaries-extra' or load 'glossaries-accsupp'\MessageBreak
    before loading 'glossaries-extra'}%
}

```

```

\fi
}
{}
\def\compatibleglossentry#1#2{%
\ toks@{#2}%
\protected@edef\do@glossentry{%
\ noexpand\accsuppglossaryentryfield{#1}%
\ noexpand\glsnamefont
\ expandafter\expandonce\csname glo@\glsdetoklabel{#1}@name\endcsname}}%
\ expandafter\expandonce\csname glo@\glsdetoklabel{#1}@desc\endcsname}%
\ expandafter\expandonce\csname glo@\glsdetoklabel{#1}@symbol\endcsname}%
\ the\toks@}%
}%
\do@glossentry
}
\def\compatiblesubglossentry#1#2#3{%
\ toks@{#3}%
\protected@edef\do@subglossentry{%
\ noexpand\accsuppglossarysubentryfield{\number#1}%
\ #2}%
\ noexpand\glsnamefont
\ expandafter\expandonce\csname glo@\glsdetoklabel{#2}@name\endcsname}}%
\ expandafter\expandonce\csname glo@\glsdetoklabel{#2}@desc\endcsname}%
\ expandafter\expandonce\csname glo@\glsdetoklabel{#2}@symbol\endcsname}%
\ the\toks@}%
}%
\do@subglossentry
}
\RequirePackage{glossaries}[v4.46]
\providecommand{\gls@accsupp@engine}{accsupp}
\providecommand{\gls@accessibility}[4]{#4}
\ifdefstring\gls@accsupp@engine{accsupp}
{
\RequirePackage{accsupp}
\renewcommand{\gls@accessibility}[4]{%
\BeginAccSupp{#1,#2={#3}}#4\EndAccSupp{}}
}
}
{}
\newcommand{\glsaccessibility}[4][[]]{%
\@glsshowaccsupp{#1}{#2}{#3}%
\gls@accessibility{#1}{#2}{#3}{#4}%
}
\define@key{glossentry}{access}{%
\def\@glo@access{#1}%
}
\define@key{glossentry}{textaccess}{%
\def\@glo@textaccess{#1}%
}
\define@key{glossentry}{firstaccess}{%

```

```

\def\@glo@firstaccess{#1}%
}
\define@key{glossentry}{pluralaccess}{%
\def\@glo@pluralaccess{#1}%
}
\define@key{glossentry}{firstpluralaccess}{%
\def\@glo@firstpluralaccess{#1}%
}
\define@key{glossentry}{symbolaccess}{%
\def\@glo@symbolaccess{#1}%
}
\define@key{glossentry}{symbolpluralaccess}{%
\def\@glo@symbolpluralaccess{#1}%
}
\define@key{glossentry}{descriptionaccess}{%
\def\@glo@descaccess{#1}%
}
\define@key{glossentry}{descriptionpluralaccess}{%
\def\@glo@descpluralaccess{#1}%
}
\define@key{glossentry}{shortaccess}{%
\def\@glo@shortaccess{#1}%
}
\define@key{glossentry}{shortpluralaccess}{%
\def\@glo@shortpluralaccess{#1}%
}
\define@key{glossentry}{longaccess}{%
\def\@glo@longaccess{#1}%
}
\define@key{glossentry}{longpluralaccess}{%
\def\@glo@longpluralaccess{#1}%
}
\define@key{glossentry}{user1access}{%
\def\@glo@useriaccess{#1}%
}
\define@key{glossentry}{user2access}{%
\def\@glo@useriiaccess{#1}%
}
\define@key{glossentry}{user3access}{%
\def\@glo@useriiiaccess{#1}%
}
\define@key{glossentry}{user4access}{%
\def\@glo@userivaccess{#1}%
}
\define@key{glossentry}{user5access}{%
\def\@glo@uservaccess{#1}%
}
\define@key{glossentry}{user6access}{%
\def\@glo@userviaccess{#1}%
}
}

```

```

\appto\@gls@keymap{,%
  {access}{access},%
  {textaccess}{textaccess},%
  {firstaccess}{firstaccess},%
  {pluralaccess}{pluralaccess},%
  {firstpluralaccess}{firstpluralaccess},%
  {symbolaccess}{symbolaccess},%
  {symbolpluralaccess}{symbolpluralaccess},%
  {descaccess}{descaccess},%
  {descpluralaccess}{descpluralaccess},%
  {shortaccess}{shortaccess},%
  {shortpluralaccess}{shortpluralaccess},%
  {longaccess}{longaccess},%
  {longpluralaccess}{longpluralaccess},%
  {user1access}{useriaccess},%
  {user2access}{useriiaccess},%
  {user3access}{useriiiaccess},%
  {user4access}{userivaccess},%
  {user5access}{uservaccess},%
  {user6access}{userviaccess}%
}
\def\@gls@noaccess{\relax}
\let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
\renewcommand*{\@newglossaryentryprehook}{%
  \@gls@oldnewglossaryentryprehook
  \def\@glo@access{\relax}%
  \def\@glo@textaccess{\@glo@access}%
  \def\@glo@firstaccess{\@glo@access}%
  \def\@glo@pluralaccess{\@glo@textaccess}%
  \def\@glo@firstpluralaccess{\@glo@pluralaccess}%
  \def\@glo@symbolaccess{\relax}%
  \def\@glo@symbolpluralaccess{\@glo@symbolaccess}%
  \def\@glo@descaccess{\relax}%
  \def\@glo@descpluralaccess{\@glo@descaccess}%
  \def\@glo@shortaccess{\relax}%
  \def\@glo@shortpluralaccess{\@glo@shortaccess}%
  \def\@glo@longaccess{\relax}%
  \def\@glo@longpluralaccess{\@glo@longaccess}%
  \def\@glo@useriaccess{\relax}%
  \def\@glo@useriiaccess{\relax}%
  \def\@glo@useriiiaccess{\relax}%
  \def\@glo@userivaccess{\relax}%
  \def\@glo@uservaccess{\relax}%
  \def\@glo@userviaccess{\relax}%
}
\let\@gls@oldnewglossaryentryposthook\@newglossaryentryposthook
\renewcommand*{\@newglossaryentryposthook}{%
  \@gls@oldnewglossaryentryposthook
  \expandafter
  \protected@xdef\csname glo@\@glo@label @access\endcsname{%

```

```

        \@glo@access}%
\expandafter
\protected@xdef\csname glo@\@glo@label @textaccess\endcsname{%
    \@glo@textaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstaccess\endcsname{%
    \@glo@firstaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @pluralaccess\endcsname{%
    \@glo@pluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstpluralaccess\endcsname{%
    \@glo@firstpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolaccess\endcsname{%
    \@glo@symbolaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolpluralaccess\endcsname{%
    \@glo@symbolpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descaccess\endcsname{%
    \@glo@descaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descpluralaccess\endcsname{%
    \@glo@descpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortaccess\endcsname{%
    \@glo@shortaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortpluralaccess\endcsname{%
    \@glo@shortpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longaccess\endcsname{%
    \@glo@longaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longpluralaccess\endcsname{%
    \@glo@longpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriaccess\endcsname{%
    \@glo@useriaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiaccess\endcsname{%
    \@glo@useriiaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiiaccess\endcsname{%
    \@glo@useriiiaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @userivaccess\endcsname{%
    \@glo@userivaccess}%
\expandafter

```

```

\protected@xdef\csname glo@\@glo@label @servaccess\endcsname{%
  \@glo@servaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @serviaccess\endcsname{%
  \@glo@serviaccess}%
}
\newcommand*\glsentryaccess}[1]{%
  \@gls@entry@field{#1}{access}%
}
\newcommand*\glsentrytextaccess}[1]{%
  \@gls@entry@field{#1}{textaccess}%
}
\newcommand*\glsentryfirstaccess}[1]{%
  \@gls@entry@field{#1}{firstaccess}%
}
\newcommand*\glsentrypluralaccess}[1]{%
  \@gls@entry@field{#1}{pluralaccess}%
}
\newcommand*\glsentryfirstpluralaccess}[1]{%
  \@gls@entry@field{#1}{firstpluralaccess}%
}
\newcommand*\glsentrysymbolaccess}[1]{%
  \@gls@entry@field{#1}{symbolaccess}%
}
\newcommand*\glsentrysymbolpluralaccess}[1]{%
  \@gls@entry@field{#1}{symbolpluralaccess}%
}
\newcommand*\glsentrydescaccess}[1]{%
  \@gls@entry@field{#1}{descaccess}%
}
\newcommand*\glsentrydescpluralaccess}[1]{%
  \@gls@entry@field{#1}{descpluralaccess}%
}
\newcommand*\glsentryshortaccess}[1]{%
  \@gls@entry@field{#1}{shortaccess}%
}
\newcommand*\glsentryshortpluralaccess}[1]{%
  \@gls@entry@field{#1}{shortpluralaccess}%
}
\newcommand*\glsentrylongaccess}[1]{%
  \@gls@entry@field{#1}{longaccess}%
}
\newcommand*\glsentrylongpluralaccess}[1]{%
  \@gls@entry@field{#1}{longpluralaccess}%
}
\newcommand*\glsentryuseriaccess}[1]{%
  \@gls@entry@field{#1}{useriaccess}%
}
\newcommand*\glsentryuseriiaccess}[1]{%
  \@gls@entry@field{#1}{useriiaccess}%
}

```

```

}
\newcommand*{\glsentryuseriiiaccess}[1]{%
  \@gls@entry@field{#1}{useriiiaccess}%
}
\newcommand*{\glsentryuserivaccess}[1]{%
  \@gls@entry@field{#1}{userivaccess}%
}
\newcommand*{\glsentryuservaccess}[1]{%
  \@gls@entry@field{#1}{uservaccess}%
}
\newcommand*{\glsentryuserviaccess}[1]{%
  \@gls@entry@field{#1}{userviaccess}%
}
\newcommand{\glsfieldaccsupp}[4]{%
  \ifdef\glscategory
  {%
    \ifcsdef{glsxtr\glscategory{#4}#3accsupp}%
    {\csname glsxtr\glscategory{#4}#3accsupp\endcsname{#1}{#2}}%
    {%
      \ifcsdef{glsxtr\glscategory{#4}accsupp}%
      {\csname glsxtr\glscategory{#4}accsupp\endcsname{#1}{#2}}%
      {%
        \ifcsdef{gls#3accsupp}%
        {\csname gls#3accsupp\endcsname{#1}{#2}}%
        {\glsaccsupp{#1}{#2}}%
      }%
    }%
  }%
  {%
    \ifcsdef{gls#3accsupp}%
    {\csname gls#3accsupp\endcsname{#1}{#2}}%
    {\glsaccsupp{#1}{#2}}%
  }%
}
\newcommand{\xglsfieldaccsupp}[1]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsfieldaccsupp\expandafter{\@gls@replacementtext}%
}
\newcommand*{\glsshortaccsupp}[2]{\glsaccessibility{E}{#1}{#2}}
\newcommand*{\glsshortplaccsupp}{\glsshortaccsupp}
\newcommand*{\glsaccsupp}[2]{\glsaccessibility{ActualText}{#1}{#2}}
\newcommand*{\xglsaccsupp}[2]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsaccsupp\expandafter{\@gls@replacementtext}{#2}%
}
\newcommand*{\@gls@access@display}[2]{%
  \protected@edef\@glo@access{#2}%
  \ifx\@glo@access\@gls@noaccess
  #1%
  \else

```

```

        \xglsaccsupp{\@glo@access}{#1}%
    \fi
}
\newcommand*\@gls@fieldaccess@display[4]{%
    \protected@edef\@glo@access{#4}%
    \ifdefequal\@glo@access\@gls@noaccess
        {#3}%
    {\expandafter\glsfieldaccsupp\expandafter{\@glo@access}{#3}{#2}{#1}}%
}
\newrobustcmd*\glsnameaccessdisplay[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@access}%
        {#1}%
    {\@gls@fieldaccess@display{#2}{name}{#1}{\glsentryaccess{#2}}}%
}
\newrobustcmd*\glsstextaccessdisplay[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*\glsymbolaccessdisplay[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@symbolaccess}%
        {#1}%
    {\@gls@fieldaccess@display{#2}{symbol}{#1}{\glsentrysymbolaccess{#2}}}%
}
\newrobustcmd*\glsymbolpluralaccessdisplay[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@symbolpluralaccess}%
        {#1}%
    {\@gls@fieldaccess@display{#2}{symbolplural}{#1}{\glsentrysymbolpluralaccess{#2}}}%
}
\newrobustcmd*\glsdescriptionaccessdisplay[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@descaccess}%
        {#1}%
    {\@gls@fieldaccess@display{#2}{desc}{#1}{\glsentrydescaccess{#2}}}%
}
\newrobustcmd*\glsdescriptionpluralaccessdisplay[2]{%

```



```

\ifcsundef{glo@glstdetoklabel{#2}@descpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{descplural}{#1}{\glstentrydescpluralaccess{#2}}}%
}
\newrobustcmd*{\glsshortaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@shortaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{short}{#1}{\glstentryshortaccess{#2}}}%
}
\newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@shortpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glstentryshortpluralaccess{#2}}}%
}
\newrobustcmd*{\glslongaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@longaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{long}{#1}{\glstentrylongaccess{#2}}}%
}
\newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@longpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{longpl}{#1}{\glstentrylongpluralaccess{#2}}}%
}
\newrobustcmd*{\glsuseriaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@useriaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useri}{#1}{\glstentryuseriaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@useriiaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{userii}{#1}{\glstentryuseriiaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiiaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@useriiiaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useriii}{#1}{\glstentryuseriiiaccess{#2}}}%
}
\newrobustcmd*{\glsuserivaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@userivaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useriv}{#1}{\glstentryuserivaccess{#2}}}%
}
\newrobustcmd*{\glsuservaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@uservaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{userv}{#1}{\glstentryuservaccess{#2}}}%
}
\newrobustcmd*{\glsuserviaccessdisplay}[2]{%

```

```

\ifcsundef{glo@\glsdetoklabel{#2}@userviaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{uservi}{#1}{\glsentryuserviaccess{#2}}}%
}
\newrobustcmd*{\glsaccessdisplay}[3]{%
\ifcsundef{gls#1accessdisplay}%
{%
\PackageError{glossaries-accsupp}{No accessibility support
for key ‘#1’}{%
}%
}%
\csname gls#1accessdisplay\endcsname{#2}{#3}%
}%
}
\renewcommand*{\@gls@default@entryfmt}[2]{%
\ifdefempty\glscustomtext
{%
\glsifplural
{%
\glscapscase
{%
\ifglsused\glslabel
{%
#2{\glspluralaccessdisplay
{\glsentryplural{\glslabel}}{\glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glsentrydescplural{\glslabel}}{\glslabel}}%
{\glsymbolpluralaccessdisplay
{\glsentrysymbolplural{\glslabel}}{\glslabel}}
{\glsinsert}}%
}%
}%
#1{\glsfirstpluralaccessdisplay
{\glsentryfirstplural{\glslabel}}{\glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glsentrydescplural{\glslabel}}{\glslabel}}%
{\glsymbolpluralaccessdisplay
{\glsentrysymbolplural{\glslabel}}{\glslabel}}%
{\glsinsert}}%
}%
}%
}%
\ifglsused\glslabel
{%
#2{\glspluralaccessdisplay
{\Glsentryplural{\glslabel}}{\glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glsentrydescplural{\glslabel}}{\glslabel}}%
{\glsymbolpluralaccessdisplay
{\glsentrysymbolplural{\glslabel}}{\glslabel}}%
}
}
}

```

```

        {\glsinsert}%
    }%
    {%
        #1{\glsfirstpluralaccessdisplay
            {\Glsentryfirstplural{\glslabel}}{\glslabel}}%
            {\glsdescriptionpluralaccessdisplay
            {\glsentrydescplural{\glslabel}}{\glslabel}}%
            {\glsymbolpluralaccessdisplay
            {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
            {\glsinsert}%
        }%
    }%
    {%
        \ifglsused\glslabel
        {%
            \MakeUppercase{%
                #2{\glspluralaccessdisplay
                    {\glsentryplural{\glslabel}}{\glslabel}}%
                    {\glsdescriptionpluralaccessdisplay
                    {\glsentrydescplural{\glslabel}}{\glslabel}}%
                    {\glsymbolpluralaccessdisplay
                    {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
                    {\glsinsert}}%
            }%
        }%
        {%
            \MakeUppercase{%
                #1{\glsfirstpluralaccessdisplay
                    {\glsentryfirstplural{\glslabel}}{\glslabel}}%
                    {\glsdescriptionpluralaccessdisplay
                    {\glsentrydescplural{\glslabel}}{\glslabel}}%
                    {\glsymbolpluralaccessdisplay
                    {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
                    {\glsinsert}}%
            }%
        }%
    }%
    {%
        \glscapscase
        {%
            \ifglsused\glslabel
            {%
                #2{\glstextaccessdisplay
                    {\glsentrytext{\glslabel}}{\glslabel}}%
                    {\glsdescriptionaccessdisplay
                    {\glsentrydesc{\glslabel}}{\glslabel}}%
                    {\glsymbolaccessdisplay
                    {\glsentrysymbol{\glslabel}}{\glslabel}}%
                    {\glsinsert}%
                }%
            }%
        }%
    }%

```

```

#1{\glsfirstaccessdisplay
  {\glsentryfirst{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glsentrydesc{\glslabel}}{\glslabel}}%
  {\glsymbolaccessdisplay
    {\glsentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
}%
{%
\ifglsused\glslabel
{%
#2{\glstextaccessdisplay
  {\Glsentrytext{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glsentrydesc{\glslabel}}{\glslabel}}%
  {\glsymbolaccessdisplay
    {\glsentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
{%
#1{\glsfirstaccessdisplay
  {\Glsentryfirst{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glsentrydesc{\glslabel}}{\glslabel}}%
  {\glsymbolaccessdisplay
    {\glsentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
}%
{%
\ifglsused\glslabel
{%
\MakeUppercase{%
#2{\glstextaccessdisplay
  {\glsentrytext{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glsentrydesc{\glslabel}}{\glslabel}}%
  {\glsymbolaccessdisplay
    {\glsentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
{%
\MakeUppercase{%
#1{\glsfirstaccessdisplay
  {\glsentryfirst{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glsentrydesc{\glslabel}}{\glslabel}}%
  {\glsymbolaccessdisplay
    {\glsentrysymbol{\glslabel}}{\glslabel}}%

```

```

        {\glsinsert}}%
    }%
} %
} %
} %
{ %
\ifglsused{\glslabel}%
{ %
    #2{\glscustomtext}%
    {\glsdescriptionaccessdisplay
     {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
     {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
} %
{ %
    #1{\glscustomtext}%
    {\glsdescriptionaccessdisplay
     {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
     {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
} %
} %
}
\renewcommand*{\glsgenentryfmt}{%
\ifdefempty\glscustomtext
{ %
    \glsifplural
    { %
        \glscapscase
        { %
            \ifglsused\glslabel
            { %
                \glspluralaccessdisplay
                {\glsentryplural{\glslabel}}{\glslabel}%
                \glsinsert
            } %
            { %
                \glsfirstpluralaccessdisplay
                {\glsentryfirstplural{\glslabel}}{\glslabel}%
                \glsinsert
            } %
        } %
    } %
    { %
        \ifglsused\glslabel
        { %
            \glspluralaccessdisplay
            {\Glsentryplural{\glslabel}}{\glslabel}%
            \glsinsert
        } %
    } %
}

```

```

}%
{%
  \glsfirstpluralaccessdisplay
    {\Glsentryfirstplural{\glslabel}}{\glslabel}%
  \glsinsert
}%
}%
{%
  \ifglsused\glslabel
  {%
    \glspluralaccessdisplay
      {\mfirstucMakeUppercase{\glsentryplural{\glslabel}}}%
      {\glslabel}%
    \mfirstucMakeUppercase{\glsinsert}%
  }%
  {%
    \glsfirstpluralaccessdisplay
      {\mfirstucMakeUppercase{\glsentryfirstplural{\glslabel}}}%
      {\glslabel}%
    \mfirstucMakeUppercase{\glsinsert}%
  }%
  }%
}%
{%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      \glstextaccessdisplay{\glsentrytext{\glslabel}}{\glslabel}%
      \glsinsert
    }%
    {%
      \glsfirstaccessdisplay{\glsentryfirst{\glslabel}}{\glslabel}%
      \glsinsert
    }%
  }%
  {%
    \ifglsused\glslabel
    {%
      \glstextaccessdisplay{\Glsentrytext{\glslabel}}{\glslabel}%
      \glsinsert
    }%
    {%
      \glsfirstaccessdisplay{\Glsentryfirst{\glslabel}}{\glslabel}%
      \glsinsert
    }%
  }%
  }%
  {%
    \ifglsused\glslabel
    {%

```

```

        \glstextaccessdisplay
          {\mfirstucMakeUppercase{\glentrytext{\glslabel}}}{\glslabel}%
        \mfirstucMakeUppercase{\glsinsert}%
      }%
    {%
      \glsfirstaccessdisplay
        {\mfirstucMakeUppercase{\glentryfirst{\glslabel}}}{\glslabel}%
      \mfirstucMakeUppercase{\glsinsert}%
    }%
  }%
}%
}%
}%
{%
  \glscustomtext\glsinsert
}%
}
\renewcommand*{\glsgenacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglsused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \acronymfont
            {\glsshortpluralaccessdisplay
              {\glentryshortpl{\glslabel}}{\glslabel}}%
          \glsinsert
        }%
      }%
      {%
        \acronymfont
          {\glsshortpluralaccessdisplay
            {\Glsentryshortpl{\glslabel}}{\glslabel}}%
          \glsinsert
        }%
      }%
      {%
        \mfirstucMakeUppercase
          {\acronymfont
            {\glsshortpluralaccessdisplay
              {\glentryshortpl{\glslabel}}{\glslabel}}%
          \glsinsert}%
      }%
    }%
  }%
  {%
    \glscapscase
    {%
      \acronymfont
        {\glsshortaccessdisplay{\glentryshort{\glslabel}}{\glslabel}}%
      \glsinsert
    }%
  }%
}

```

```

    }%
    {%
      \acronymfont
      {\glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
      \glsinsert
    }%
    {%
      \mfirstucMakeUppercase
      {\acronymfont{%
        \glsshortaccessdisplay{\glsentryshort{\glslabel}}{\glslabel}}%
        \glsinsert}%
    }%
  }%
}
{%
\glsifplural
{%
  \glscapscase
  {%
    \genplacrfullformat{\glslabel}{\glsinsert}%
  }%
  {%
    \Genplacrfullformat{\glslabel}{\glsinsert}%
  }%
  {%
    \mfirstucMakeUppercase
    {\genplacrfullformat{\glslabel}{\glsinsert}}%
  }%
}
{%
  \glscapscase
  {%
    \genacrfullformat{\glslabel}{\glsinsert}%
  }%
  {%
    \Genacrfullformat{\glslabel}{\glsinsert}%
  }%
  {%
    \mfirstucMakeUppercase
    {\genacrfullformat{\glslabel}{\glsinsert}}%
  }%
}
}
}
\renewcommand*{\genacrfullformat}[2]{%
  \glslongaccessdisplay{\glsentrylong{#1}}{#1}#2\space

```



```

(\glsshortaccessdisplay{\protect\firstacronymfont{\glentryshort{#1}}}{#1})%
}
\renewcommand*{\Genacrfullformat}[2]{%
\glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space
(\glsshortaccessdisplay{\protect\firstacronymfont{\glentryshort{#1}}}{#1})%
}
\renewcommand*{\genplacrfullformat}[2]{%
\glslongpluralaccessdisplay{\glentrylongpl{#1}}{#1}#2\space
(\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glentryshortpl{#1}}}{#1})%
}
\renewcommand*{\Genplacrfullformat}[2]{%
\glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}#2\space
(\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glentryshortpl{#1}}}{#1})%
}
\def\@acrshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glsshortaccessdisplay{\glentryshort{#2}}{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@Acrshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@secondofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@ACRshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@thirdofthree

```

```

\let\glsinsert\@empty
\def\glscustomtext{%
  \acronymfont{\glsshortaccessdisplay
    {\MakeUppercase{\glsentryshort{#2}}}{#2}}#3%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{\glsentrylong{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@AcrLong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{\Glsentrylong{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@ACRLong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{%
        \MakeUppercase{\glsentrylong{#2}}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%

```

```

\glspostlinkhook
}
\def\@glstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\glstextentrytext{#2}}{#2}#3}%
}
\def\@GLstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\GLstextentrytext{#2}}{#2}#3}%
}
\def\@GLStext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glstextaccessdisplay{\mfirstucMakeUppercase{\glstextentrytext{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\glstextentryfirst{#2}}{#2}#3}%
}
\def\@Glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\GLstextentryfirst{#2}}{#2}#3}%
}
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsfirstaccessdisplay{\mfirstucMakeUppercase{\glstextentryfirst{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\glstextentryplural{#2}}{#2}#3}%
}
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\GLstextentryplural{#2}}{#2}#3}%
}
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glspluralaccessdisplay{\mfirstucMakeUppercase{\glstextentryplural{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\glstextentryfirstplural{#2}}{#2}#3}%
}
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\GLstextentryfirstplural{#2}}{#2}#3}%
}
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsfirstpluralaccessdisplay{\mfirstucMakeUppercase{\glstextentryfirstplural{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glstextentryname{#2}}{#2}#3}%
}
\def\@Glsname@#1#2[#3]{%

```

```

\@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\Glsentryname{#2}}{#2}#3}%
}
\def\@GLSname@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsnameaccessdisplay{\mfirstucMakeUppercase{\glsentryname{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@glsdesc@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\glsentrydesc{#2}}{#2}#3}%
}
\def\@Glsdesc@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
}
\def\@GLSdesc@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsdescriptionaccessdisplay{\mfirstucMakeUppercase{\glsentrydesc{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@glsdescplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\glsentrydescplural{#2}}{#2}#3}%
}
\def\@Glsdescplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
}
\def\@GLSdescplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsdescriptionpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrydescplural{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@glsymbol@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
}
\def\@Glsymbol@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}
\def\@GLSsymbol@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsymbolaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbol{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@glsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsymbolpluralaccessdisplay{\glsentrysymbolplural{#2}}{#2}#3}%
}
\def\@Glsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsymbolpluralaccessdisplay{\Glsentrysymbolplural{#2}}{#2}#3}%
}
\def\@GLSsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsymbolpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbolplural{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}

```

```

}
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glsentryuseri{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriaccessdisplay{\mfirstucMakeUppercase{\glsentryuseri{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glsentryuserii{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glsentryuserii{#2}}{#2}#3}%
}
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiaccessdisplay{\mfirstucMakeUppercase{\glsentryuserii{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\glsentryuseriii{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glsentryuseriii{#2}}{#2}#3}%
}
\def\@GLSuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiiaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriii{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glsentryuseriv{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
}
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuserivaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriv{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\glsentryuserv{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\Glsentryuserv{#2}}{#2}#3}%
}

```

```

}
\def\@GLSuser@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuseraccessdisplay{\mfirstucMakeUppercase{\glsentryuser{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glsentryuseri{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuseriaccessdisplay{\mfirstucMakeUppercase{\glsentryuseri{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
}
\renewcommand*\@glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\glsentryname{#1}}{#1}}%
  }%
}
\renewcommand*\@glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\Glsentryname{#1}}{#1}}%
  }%
}
\renewcommand*\@glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\glsentrydesc{#1}}{#1}%
  }%
}
\renewcommand*\@Glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}
\renewcommand*\@glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsymbolaccessdisplay{\glsentrysymbol{#1}}{#1}%
  }%
}
\renewcommand*\@Glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%

```

```

        \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
    }%
}
\newcommand*\accsuppglossaryentryfield}[5]{%
    \glossaryentryfield{#1}%
    {\glsnameaccessdisplay{#2}{#1}}%
    {\glsdescriptionaccessdisplay{#3}{#1}}%
    {\glssymbolaccessdisplay{#4}{#1}}{#5}%
}
\newcommand*\accsuppglossarysubentryfield}[6]{%
    \glossarysubentryfield{#1}{#2}%
    {\glsnameaccessdisplay{#3}{#2}}%
    {\glsdescriptionaccessdisplay{#4}{#2}}%
    {\glssymbolaccessdisplay{#5}{#2}}{#6}%
}
\renewacronymstyle{long-short}%
{%
    \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
    \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
    \renewcommand*\genacrfullformat}[2]{%
        \glslongaccessdisplay{\glsentrylong{##1}}{##1}##2\space
        (\glsshortaccessdisplay
            {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
    }%
    \renewcommand*\Genacrfullformat}[2]{%
        \glslongaccessdisplay{\Glsentrylong{##1}}{##1}##2\space
        (\glsshortaccessdisplay
            {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
    }%
    \renewcommand*\genplacrfullformat}[2]{%
        \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}##2\space
        (\glsshortpluralaccessdisplay
            {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
    }%
    \renewcommand*\Genplacrfullformat}[2]{%
        \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}##2\space
        (\glsshortpluralaccessdisplay
            {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
    }%
    \renewcommand*\acronymentry}[1]{%
        \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}
    }
    \renewcommand*\acronymsort}[2]{##1}%
    \renewcommand*\acronymfont}[1]{##1}%
    \renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
    \renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\renewacronymstyle{short-long}%
{%

```

```

\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2\space
(\glslongaccessdisplay{\glsentrylong{##1}}{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\Glsentryshort{##1}}}{##1}##2\space
(\glslongaccessdisplay{\glsentrylong{##1}}{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2\space
(\glslongpluralaccessdisplay
{\glsentrylongpl{##1}}{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2\space
(\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\renewacronymstyle{long-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}
\renewacronymstyle{long-sc-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sc-short}%
}

```



```

\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%

```

```

\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glslongpluralaccessdisplay{\glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay{\Glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay
            {\mfirstucMakeUppercase{\glsentrylongpl{\glslabel}}}{\glslabel}%
            \mfirstucMakeUppercase{\glsinsert}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \glslongaccessdisplay{\glsentrylong{\glslabel}}{\glslabel}\glsinsert
      }%
      {%
        \glslongaccessdisplay{\Glsentrylong{\glslabel}}{\glslabel}\glsinsert
      }%
      {%
        \glslongaccessdisplay
          {\mfirstucMakeUppercase
            {\glsentrylong{\glslabel}\glsinsert}}{\glslabel}%
          \mfirstucMakeUppercase{\glsinsert}%
      }%
    }%
  }%
  {%
    \glsgenentryfmt
  }%
  {\glscustomtext\glsinsert}%
}%
{%

```

```

\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay
    {\mfirstucMakeUppercase{\glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\Glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\mfirstucMakeUppercase{\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\glsentryfull}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
  \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}\space
  (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space
  (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}%
}

```

```

\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\renewacronymstyle{dua-desc}%
{%
\GlsUseAcrEntryDispStyle{dua}%
}%
{%
\GlsUseAcrStyleDefs{dua}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymentry}[1]{%
\glslongaccessdisplay{\acronymfont{\glsentrylong{##1}}}{##1}}%
\renewcommand*\acronymsort}[2]{##2}%
}%
\renewacronymstyle{footnote}%
{%
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*\genacrfullformat}[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}}{##1}}%
}%
\renewcommand*\Genacrfullformat}[2]{%
\glsshortaccessdisplay
{\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}}{##1}}%
}%
\renewcommand*\genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%
}%
\renewcommand*\Genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%
}%
\renewcommand*\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
\renewcommand*\acrfullfmt}[3]{%
\glslink[##1]{##2}{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}##3\space

```

```

        (\glslongaccessdisplay{\glsentrylong{##2}{##2}})}%
\renewcommand*\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##2}}}{##2}##3\space
        (\glslongaccessdisplay{\glsentrylong{##2}{##2}})}%
\renewcommand*\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortaccessdisplay
        {\mfirstucMakeUppercase
         {\acronymfont{\glsentryshort{##2}}}{##2}##3\space
         (\glslongaccessdisplay{\glsentrylong{##2}{##2}})}}%
\renewcommand*\acrfullplfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortpluralaccessdisplay
        {\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
        (\glslongpluralaccessdisplay{\glsentrylongpl{##2}{##2}})}%
\renewcommand*\Acrfullplfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortpluralaccessdisplay
        {\acronymfont{\Glsentryshortpl{##2}}}{##2}##3\space
        (\glslongpluralaccessdisplay{\glsentrylongpl{##2}})}%
\renewcommand*\ACRfullplfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortpluralaccessdisplay
        {\mfirstucMakeUppercase
         {\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
         (\glslongpluralaccessdisplay{\glsentrylongpl{##2}{##2}})}}%
\renewcommand*\glsentryfull}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}\space
    (\glslongaccessdisplay{\glsentrylong{##1}{##1}})%
\renewcommand*\Glsentryfull}[1]{%
    \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##1}}}{##1}\space
    (\glslongaccessdisplay{\glsentrylong{##1}{##1}})%
\renewcommand*\glsentryfullpl}[1]{%
    \glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##1}}}{##1}\space
    (\glslongpluralaccessdisplay{\glsentrylongpl{##1}{##1}})%
\renewcommand*\Glsentryfullpl}[1]{%
    \glsshortpluralaccessdisplay
    {\acronymfont{\Glsentryshortpl{##1}}}{##1}\space
    (\glslongpluralaccessdisplay{\glsentrylongpl{##1}{##1}})%
}
\renewacronymstyle{footnote-sc}%
{%
    \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
    \GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}

```

```

\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glstextup{\glspluralsuffix}}%
}%
\renewacronymstyle{footnote-sm}%
{%
\GlsUseAcrEntryDispStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}%
\renewacronymstyle{footnote-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sc-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
\GlsUseAcrStyleDefs{footnote-sc}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sm-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote-sm}%
}%
{%
\GlsUseAcrStyleDefs{footnote-sm}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
}

```

```

\newcommand*\glsdefaultshortaccess}[2]{#1}
\renewcommand*\newacronymhook{%
  \edef\@gls@keylist{%
    shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    shortpluralaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    \the\glskeylisttok}%
  \expandafter\glskeylisttok\expandafter{\@gls@keylist}%
}
\renewcommand*\DefaultNewAcronymDef{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      description={\the\glslongtok},%
      descriptionaccess=\relax,
      text={\the\glsshorttok},%
      access={\noexpand\@glo@textaccess},%
      sort={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      first={\noexpand\glslongaccessdisplay
        {\the\glslongtok}{\the\glslabeltok}\space
        (\noexpand\glsshortaccessdisplay
          {\the\glsshorttok}{\the\glslabeltok})},%
      plural={\the\glsshorttok\acrpluralsuffix},%
      firstplural={\noexpand\glslongpluralaccessdisplay
        {\noexpand\@glo@longpl}{\the\glslabeltok}\space
        (\noexpand\glsshortpluralaccessdisplay
          {\noexpand\@glo@shortpl}{\the\glslabeltok})},%
      firstaccess=\relax,
      firstpluralaccess=\relax,
      textaccess={\noexpand\@glo@shortaccess},%
      \the\glskeylisttok
    }%
  }%
  \let\@org@gls@assign@firstpl\gls@assign@firstpl
  \let\@org@gls@assign@plural\gls@assign@plural
  \let\@org@gls@assign@descplural\gls@assign@descplural
  \def\gls@assign@firstpl##1##2{%
    \@gls@expand@field{##1}{firstpl}{##2}%
  }%
  \def\gls@assign@plural##1##2{%
    \@gls@expand@field{##1}{plural}{##2}%
  }%
  \def\gls@assign@descplural##1##2{%

```

```

    \@gls@expand@field{##1}{descplural}{##2}%
  }%
  \@do@newglossaryentry
  \let\gls@assign@firstpl\@org@gls@assign@firstpl
  \let\gls@assign@plural\@org@gls@assign@plural
  \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\DescriptionFootnoteNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\noexpand\acronymfont{\the\glsshorttok}},%
      sort={\the\glsshorttok},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      access={\noexpand\@glo@textaccess},%
      plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      symbol={\the\glslongtok},%
      symbolplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      firstpluralaccess=\relax,
      textaccess={\noexpand\@glo@shortaccess},%
      \the\glskeylisttok
    }%
  }%
  \let\@org@gls@assign@firstpl\gls@assign@firstpl
  \let\@org@gls@assign@plural\gls@assign@plural
  \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
  \def\gls@assign@firstpl##1##2{%
    \@gls@expand@field{##1}{firstpl}{##2}%
  }%
  \def\gls@assign@plural##1##2{%
    \@gls@expand@field{##1}{plural}{##2}%
  }%
  \def\gls@assign@symbolplural##1##2{%
    \@gls@expand@field{##1}{symbolplural}{##2}%
  }%
  \@do@newglossaryentry
  \let\gls@assign@plural\@org@gls@assign@plural
  \let\gls@assign@firstpl\@org@gls@assign@firstpl
  \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\DescriptionNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%

```



```

type=\acronymtype,%
name={\noexpand
  \acronymformat{\the\glsshorttok}{\the\glslongtok}},%
access={\noexpand\@glo@textaccess},%
sort={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
first={\the\glslongtok},%
firstaccess=\relax,
firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
text={\the\glsshorttok},%
textaccess={\the\glslongtok},%
plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
symbol={\noexpand\@glo@text},%
symbolaccess={\noexpand\@glo@textaccess},%
symbolplural={\noexpand\@glo@plural},%
firstpluralaccess=\relax,
textaccess={\noexpand\@glo@shortaccess},%
\the\glskeylisttok}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\FootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\glsshorttok}},%
    sort={\the\glsshorttok},%
    text={\the\glsshorttok},%
    textaccess={\the\glslongtok},%
    access={\noexpand\@glo@textaccess},%

```

```

plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\the\glslongtok},%
descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
\the\glskeylisttok
}%
}%
\let\org@gls@assign@plural@gls@assign@plural
\let\org@gls@assign@firstpl@gls@assign@firstpl
\let\org@gls@assign@descplural@gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@gls@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\org@gls@assign@plural
\let\gls@assign@firstpl\org@gls@assign@firstpl
\let\gls@assign@descplural\org@gls@assign@descplural
}
\renewcommand*\SmallNewAcronymDef{%
\edef\do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\glsshorttok}},%
    access={\noexpand\@glo@symbolaccess},%
    sort={\the\glsshorttok},%
    short={\the\glsshorttok},%
    shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
    shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    long={\the\glslongtok},%
    longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    text={\noexpand\@glo@short},%
    textaccess={\noexpand\@glo@shortaccess},%
    plural={\noexpand\@glo@shortpl},%
    first={\the\glslongtok},%
    firstaccess=\relax,
    firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    description={\noexpand\@glo@first},%
    descriptionplural={\noexpand\@glo@firstplural},%
    symbol={\the\glsshorttok},%
    symbolaccess={\the\glslongtok},%

```

```

        symbolplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
        \the\glskeylisttok
    }%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
    \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
    \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
    \@@gls@expand@field{##1}{descplural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
    \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@descplural\@org@gls@assign@descplural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\glsshortaccesskey}{\glsshortkey access}%
\newcommand*{\glsshortpluralaccesskey}{\glsshortpluralkey access}%
\newcommand*{\glslongaccesskey}{\glslongkey access}%
\newcommand*{\glslongpluralaccesskey}{\glslongpluralkey access}%
\newcommand*{\showglongnameaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@access\endcsname
}
\newcommand*{\showglongtextaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@textaccess\endcsname
}
\newcommand*{\showglongpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@pluralaccess\endcsname
}
\newcommand*{\showglongfirstaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@firstaccess\endcsname
}
\newcommand*{\showglongfirstpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpluralaccess\endcsname
}
\newcommand*{\showglongsymbolaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolaccess\endcsname
}
\newcommand*{\showglongsymbolpluralaccess}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolpluralaccess\endcsname
}

```

```

}
\newcommand*\showglodescaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@descaccess\endcsname
}
\newcommand*\showglodescpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@descpluralaccess\endcsname
}
\newcommand*\showgloshortaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@shortaccess\endcsname
}
\newcommand*\showgloshortpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@shortpluralaccess\endcsname
}
\newcommand*\showglolongaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@longaccess\endcsname
}
\newcommand*\showglolongpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@longpluralaccess\endcsname
}
}

```

8.3 Rollback v4.46 (glossaries-babel-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-babel}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
  }%
}
}
}

```

8.4 Rollback v4.46 (glossaries-compatible-207-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-207}[2020/03/19 v4.46 (NLCT)]
\ifglxindy
  \renewcommand*\GlsAddXdyAttribute[1]{%
    \edef\xdyattributes{\xdyattributes ^^J \string"#1\string"}%
    \expandafter\toks@\expandafter{\xdylocref}%
    \edef\xdylocref{\the\toks@ ^^J%
    (markup-locref
    :open \string"\string~\n\string\setentrycounter
      {\noexpand\glscounter}%
      \expandafter\string\csname#1\endcsname
      \expandafter@gobble\string\{\string" ^^J
    :close \string"\expandafter@gobble\string\}\string" ^^J
    :attr \string"#1\string")}}
  \fi
\renewcommand*\GlsAddXdyCounters[1]{%
  \GlossariesWarning{\string\GlsAddXdyCounters\space not available
    in compatibility mode.}%
}
\GlsAddXdyAttribute{glsnumberformat}
\GlsAddXdyAttribute{textrm}
\GlsAddXdyAttribute{textsf}
\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glsnumber}
\GlsAddXdyAttribute{hyperm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hypertt}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}
\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}
\ifglxindy
  \renewcommand*\GlsAddXdyLocation[2]{%
    \edef\xdyuserlocationdefs{%
      \xdyuserlocationdefs ^^J%
      (define-location-class \string"#1\string"^^J\space\space
      \space(#2))
  }

```

```

    }%
    \edef\xdyuserlocationnames{%
        \xdyuserlocationnames^^J\space\space\space
        \string"#1\string"}%
}
\fi
\renewcommand{\do@wrglossary}[1]{%
\ifglxsindy
    \expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
    \def\@glo@range{%
    \expandafter\if\@glo@prefix(\relax
        \def\@glo@range{:open-range}%
    \else
        \expandafter\if\@glo@prefix)\relax
        \def\@glo@range{:close-range}%
    \fi
    \fi
    \protected@edef\@glslocref{\theglentrycounter}%
    \@gls@checkmkidxchars\@glslocref
    \glossary[\csname glo@#1@type\endcsname]{%
    (indexentry :tkey (\csname glo@#1@index\endcsname)
        :locref \string"\@glslocref\string" %
        :attr \string"\@glo@suffix\string" \@glo@range
    )
    }%
\else
    \@set@glo@numformat\@glo@numfmt\@gls@counter\@glsnumberformat
    \glossary[\csname glo@#1@type\endcsname]{%
    \string\glossaryentry{\csname glo@#1@index\endcsname
        \@gls@encapchar\@glo@numfmt}{\theglentrycounter}}%
\fi
}
\def\@set@glo@numformat#1#2#3{%
    \expandafter\@glo@check@mkidxrangechar#3\@nil
    \protected@edef#1{%
        \@glo@prefix setentrycounter[]#{#2}%
        \expandafter\string\csname\@glo@suffix\endcsname
    }%
    \@gls@checkmkidxchars#1%
}
\ifglxsindy
\def\writeist{%
    \openout\glswrite=\listfilename
    \write\glswrite{;; xindy style file created by the glossaries
        package in compatible-2.07 mode}%
    \write\glswrite{;; for document '\jobname' on
        \the\year-\the\month-\the\day}%
    \write\glswrite{^^J; required styles^^J}
    \@for\xdstyle:=\xdyrequiredstyles\do{%
        \if\xdstyle\@empty

```

```

\else
  \protected@write\glswrite{}{(require
    \string"\@xdystyle.xdy\string")}%
\fi
}%
\write\glswrite{^^J%
  ; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes (\@xdyattributes))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%
\protected@edef\@gls@roman{\@roman{0}\string"
  \string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@gls@roman
\edef\@tmp{\string" \string"roman-numbers-lowercase\string"
  :sep \string"}%
\@onelevel@sanitize\@tmp
\ifx\@tmp\@gls@roman
  \write\glswrite{(define-location-class
    \string"roman-page-numbers\string"^^J\space\space\space
    (\string"roman-numbers-lowercase\string")
    :min-range-length \@glsminrange)}%
\else
  \write\glswrite{(define-location-class
    \string"roman-page-numbers\string"^^J\space\space\space
    (:sep "\@gls@roman")
    :min-range-length \@glsminrange)}%
\fi
\write\glswrite{(define-location-class
  \string"Roman-page-numbers\string"^^J\space\space\space
  (\string"roman-numbers-uppercase\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"arabic-page-numbers\string"^^J\space\space\space
  (\string"arabic-numbers\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"alpha-page-numbers\string"^^J\space\space\space
  (\string"alpha\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"Alpha-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"Appendix-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string"
  :sep \string"\@glsAlphacompositor\string"
  \string"arabic-numbers\string")
  :min-range-length \@glsminrange)}%

```

```

\write\glswrite{(define-location-class
  \string"arabic-section-numbers\string"^^J\space\space\space
  (\string"arabic-numbers\string"
   :sep \string"\glscompositor\string"
   \string"arabic-numbers\string")
   :min-range-length \@glsminrange)}}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\glsseeformat\string"
  :close \string"{}\string")}%
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
  :open \string"\string
  \glossarysection[\string\glossarytoctitle]{\string
  \glossarytitle}\string\glossarypreamble\string~n\string\begin
  {theglossary}\string\glossaryheader\string~n\string" ^^J\space
  \space\space:close \string"\expandafter@gobble
  \string~%\string~n\string
  \end{theglossary}\string\glossarypostamble
  \string~n\string" ^^J\space\space\space
  :tree)}}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\string~n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \string~n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list
  :sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
  :sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixF" :length 1 :ignore-end)}%
\fi

```



```

\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{(markup-range
:close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
:sep \string\string\glsgroupskip\string~n\string)}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
:open-head \string\string\glsgroupheading
\glsopenbrace\string^^J\space\space\space
:close-head \string\glsclosebrace\string)}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}%
\write\glswrite{\@xdysortrules}%
\noist}
\else
\edef\gls@actualchar{\string?}
\edef\gls@encapchar{\string|}
\edef\gls@levelchar{\string!}
\edef\gls@quotechar{\string"}
\def\writeist{\relax
\openout\glswrite=\istfilename
\write\glswrite{\expandafter\@gobble\string\% makeindex style file
created by the glossaries package}
\write\glswrite{\expandafter\@gobble\string\% for document
'\jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\gls@actualchar'}
\write\glswrite{encap '\gls@encapchar'}
\write\glswrite{level '\gls@levelchar'}
\write\glswrite{quote '\gls@quotechar'}
\write\glswrite{keyword \string\string\glossaryentry\string"}
\write\glswrite{preamble \string\string\glossarysection[\string
\glossarytoctitle]{\string\glossarytitle}\string
\glossarypreamble\string\n\string\begin{theglossary}\string
\glossaryheader\string\n\string"}
\write\glswrite{postamble \string\string\string\n\string
\end{theglossary}\string\glossarypostamble\string\n
\string"}
\write\glswrite{group_skip \string\string\glsgroupskip\string\n
\string"}
\write\glswrite{item_0 \string\string\string\n\string"}
\write\glswrite{item_1 \string\string\string\n\string"}
\write\glswrite{item_2 \string\string\string\n\string"}
\write\glswrite{item_01 \string\string\string\n\string"}
\write\glswrite{item_x1

```

```

\string\string\relax \string\glsresetentrylist\string\n
\string}
\write\glswrite{item_12 \string\string\%\string\n\string}
\write\glswrite{item_x2
\string\string\relax \string\glsresetentrylist\string\n
\string}
\write\glswrite{delim_0 \string\string\{\string
\glossaryentrynumbers\string\{\string\relax \string}
\write\glswrite{delim_1 \string\string\{\string
\glossaryentrynumbers\string\{\string\relax \string}
\write\glswrite{delim_2 \string\string\{\string
\glossaryentrynumbers\string\{\string\relax \string}
\write\glswrite{delim_t \string\string\}\string}\string}
\write\glswrite{delim_n \string\string\delimN \string}
\write\glswrite{delim_r \string\string\delimR \string}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
\string\string\glsgroupheading\string\{\string}
\write\glswrite{heading_suffix
\string\string\}\string\relax
\string\glsresetentrylist \string}
\write\glswrite{symhead_positive \string"glssymbols\string}
\write\glswrite{numhead_positive \string"glnumbers\string}
\write\glswrite{page_compositor \string"glscompositor\string}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{suffix_2p \string"\gls@suffixF\string}
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{suffix_3p \string"\gls@suffixFF\string}
\fi
\noist
}
\fi
\renewcommand*{\noist}{\let\writeist\relax}

```

8.5 Rollback v4.46 (glossaries-compatible-307-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-307}[2020/03/19 v4.46 (NLCT)]
\newcommand{\compatglossarystyle}[2]{%
\ifcsundef{@glscompstyle@#1}%
{%
\csdef{@glscompstyle@#1}{#2}%
}%

```

```

    {%
      \PackageError{glossaries}{Glossary compatibility style ‘#1’ is already defined}{}%
    }%
  }
\compatglossarystyle{inline}{%
  \renewcommand{\glossaryentryfield}[5]{%
    \glsinlinedopostchild
    \gls@inlinesep
    \def\glo@desc{##3}%
    \def\@no@post@desc{\nopostdesc}%
    \glsentryitem{##1}\glsinlinenameformat{##1}{##2}%
    \ifx\glo@desc\@no@post@desc
      \glsinlineemptydescformat{##4}{##5}%
    \else
      \ifstrempy{##3}%
        {\glsinlineemptydescformat{##4}{##5}}%
        {\glsinlinedescformat{##3}{##4}{##5}}%
      \fi
      \ifglshaschildren{##1}%
        {%
          \glsresetsubentrycounter
          \glsinlineparentchildseparator
          \def\gls@inlinesubsep{}%
          \def\gls@inlinepostchild{\glsinlinepostchild}%
        }%
        {}%
      \def\gls@inlinesep{\glsinlineseparator}%
    }%
  \renewcommand{\glossarysubentryfield}[6]{%
    \gls@inlinesubsep%
    \glsinlinesubnameformat{##2}{##3}%
    \glssubentryitem{##2}\glsinlinesubdescformat{##4}{##5}{##6}%
    \def\gls@inlinesubsep{\glsinlinesubseparator}%
  }%
}
\compatglossarystyle{list}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \item[\glsentryitem{##1}\glstarget{##1}{##2}]
      ##3\glspostdescription\space ##5}%
  \renewcommand*\glossarysubentryfield}[6]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6.}%
}
\compatglossarystyle{listgroup}{%
  \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{listhypergroup}{%
  \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{altlist}{%

```

```

\renewcommand*\glossaryentryfield}[5]{%
  \item[\glentryitem{##1}\glstarget{##1}{##2}]%
  \mbox{}\par\nobreak\@afterheading
  ##3\glspostdescription\space ##5}%
\renewcommand*\glossarysubentryfield}[6]{%
  \par
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}##4\glspostdescription\space ##6}%
}%
\compatglossarystyle{altlistgroup}{%
  \csuse{@glscompstyle@altlist}%
}%
\compatglossarystyle{altlisthypergroup}{%
  \csuse{@glscompstyle@altlist}%
}%
\compatglossarystyle{listdotted}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \item[]\makebox[\glslistdottedwidth][l]{%
      \glentryitem{##1}\glstarget{##1}{##2}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##3}%
  \renewcommand*\glossarysubentryfield}[6]{%
    \item[]\makebox[\glslistdottedwidth][l]{%
      \glssubentryitem{##2}%
      \glstarget{##2}{##3}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##4}%
  }%
\compatglossarystyle{sublistdotted}{%
  \csuse{@glscompstyle@listdotted}%
  \renewcommand*\glossaryentryfield}[5]{%
    \item[\glentryitem{##1}\glstarget{##1}{##2}]}%
}%
\compatglossarystyle{long}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
  }%
\compatglossarystyle{longborder}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheader}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheaderborder}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{long3col}{%
  \renewcommand*\glossaryentryfield}[5]{%

```

```

        \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]{%
        \glentryitem{##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{@glscompstyle@tree}%
}%
\compatglossarystyle{treehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{treenoname}{%
\renewcommand{\glossaryentryfield}[5]{%
\hangindent0pt\relax
\parindent0pt\relax
\glstentryitem{##1}\textbf{\glstarget{##1}{##2}}%
\ifx\relax##4\relax
\else
\space(##4)%
\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\glstarget{##2}{\strut}%
##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treenonamegroup}{%
\csuse{@glscompstyle@treenoname}%
}%
\compatglossarystyle{treenonamehypergroup}{%
\csuse{@glscompstyle@treenoname}%
}%
\compatglossarystyle{almtree}{%
\renewcommand{\glossaryentryfield}[5]{%
\ifnum\@gls@prevlevel=0\relax
\else
\settothewidth{\glstreeindent}{\textbf{\@glswidestname\space}}%
\hangindent\glstreeindent
\parindent\glstreeindent
\fi
\makebox[Opt][r]{\makebox[\glstreeindent][l]{%
\glstentryitem{##1}\textbf{\glstarget{##1}{##2}}}}%
\ifx\relax##4\relax
\else
(##4)\space
\fi
##3\glspostdescription \space ##5\par
\def\@gls@prevlevel{0}%

```



```

}%
\renewcommand{\glossarysubentryfield}[6]{%
  \ifnum##1=1\relax
    \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]{%
\glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4\glspostdescription\space ##6%
\tabularnewline}%
}%
\compatglossarystyle{superraggedborder}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheader}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheaderborder}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superragged3col}{%

```

```

\renewcommand*\glossaryentryfield}[5]{%
  \glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
\renewcommand*\glossarysubentryfield}[6]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{superragged3colborder}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheader}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheaderborder}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{altsuperragged4col}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%
\compatglossarystyle{altsuperragged4colheader}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colborder}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colheaderborder}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{super}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
}%
\compatglossarystyle{superborder}{%
  \csuse{@glscompstyle@super}%
}%
\compatglossarystyle{superheader}{%
  \csuse{@glscompstyle@super}%
}%
\compatglossarystyle{superheaderborder}{%
  \csuse{@glscompstyle@super}%
}%

```

```

\compatglossarystyle{super3col}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\\}%
}%
\compatglossarystyle{super3colborder}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super3colheader}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super3colheaderborder}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super4col}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
}%
\compatglossarystyle{super4colheader}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{super4colborder}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{super4colheaderborder}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4col}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colheader}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colborder}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colheaderborder}{%
  \csuse{@glscompstyle@super4col}%
}%

```

8.6 Rollback v4.46 (glossaries-polyglossia-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-polyglossia}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \PackageWarningNoLine{glossaries}%
      {No language module detected for ‘\this@dialect’.\MessageBreak
      Language modules need to be installed separately.\MessageBreak
      Please check on CTAN for a bundle called\MessageBreak
      ‘glossaries-\CurrentTrackedLanguage’ or similar}%
    }%
  }%
}%
}
```

8.7 Rollback v4.46 (glossaries-prefix-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-prefix}[2020/03/19 v4.46 (NLCT)]
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
\ProcessOptions
\RequirePackage{glossaries}[v4.46]
\define@key{glossentry}{prefixfirst}{\def\@glo@entryprefixfirst{#1}}%
\define@key{glossentry}{prefixfirstplural}{\def\@glo@entryprefixfirstplural{#1}}%
\define@key{glossentry}{prefix}{\def\@glo@entryprefix{#1}}%
\define@key{glossentry}{prefixplural}{\def\@glo@entryprefixplural{#1}}%
\appto\@gls@keymap{,%
  {prefixfirst}{prefixfirst},%
  {prefixfirstplural}{prefixfirstplural},%
  {prefix}{prefix},%
  {prefixplural}{prefixplural}%
}
\appto\@newglossaryentryprehook{%
  \def\@glo@entryprefix{}%
  \def\@glo@entryprefixplural{}%
  \let\@glo@entryprefixfirst\@gls@default@value
```

```

\let\@glo@entryprefixfirstplural\@gls@default@value
}
\appto\@newglossaryentryposthook{%
\gls@assign@field{}\@glo@label}{prefix}\@glo@entryprefix}%
\gls@assign@field{}\@glo@label}{prefixplural}\@glo@entryprefixplural}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @prefix\endcsname}\@glo@label}{prefixfirst}%
  {\@glo@entryprefixfirst}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @prefixplural\endcsname}\@glo@label}%
  {prefixfirstplural}\@glo@entryprefixfirstplural}%
}
\newcommand*\@glsentryprefixfirst}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixfirst}}
\newcommand*\@glsentryprefixfirstplural}[1]{%
\csuse{glo@\glsdetoklabel{#1}@prefixfirstplural}}
\newcommand*\@glsentryprefix}[1]{\csuse{glo@\glsdetoklabel{#1}@prefix}}
\newcommand*\@glsentryprefixplural}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixplural}}
\newrobustcmd*\@Glsentryprefixfirst}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixfirst\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\@Glsentryprefixfirstplural}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixfirstplural\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\@Glsentryprefix}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefix\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\@Glsentryprefixplural}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixplural\endcsname}%
\xmakefirstuc\@glo@text
}
\newcommand*\@ifglshasprefix}[3]{%
\ifcsemtyp{glo@\glsdetoklabel{#1}@prefix}%
{#3}%
{#2}%
}
\newcommand*\@ifglshasprefixplural}[3]{%
\ifcsemtyp{glo@\glsdetoklabel{#1}@prefixplural}%
{#3}%
{#2}%
}
\newcommand*\@ifglshasprefixfirst}[3]{%
\ifcsemtyp{glo@\glsdetoklabel{#1}@prefixfirst}%
{#3}%
{#2}%
}
\newcommand*\@ifglshasprefixfirstplural}[3]{%
\ifcsemtyp{glo@\glsdetoklabel{#1}@prefixfirstplural}%

```

```

    {#3}%
    {#2}%
}
\providecommand{\@glsprefix@record@hook}[2]{%
  \ifdef\@glsxtr@record
    {\@glsxtr@record{#1}{#2}{glslink}}%
  }%
}
\newcommand{\glsprefixsep}{-}
\newrobustcmd{\pgls}{\@gls@hyp@opt\@pgls}
\newcommand*{\@pgls}[2][{}]{%
  \new@ifnextchar[%
    {\@pgls@{#1}{#2}}%
    {\@pgls@{#1}{#2}[{}]}%
}
\def\@pgls@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglschasprefix{#2}{\glsentryprefix{#2}\glsprefixsep}{-}%
    }%
    {%
      \ifglschasprefixfirst{#2}{\glsentryprefixfirst{#2}\glsprefixsep}{-}%
    }%
    \@gls@{#1}{#2}[#3]%
  }%
}
\newrobustcmd{\pglsp1}{\@gls@hyp@opt\@pglsp1}
\newcommand*{\@pglsp1}[2][{}]{%
  \new@ifnextchar[%
    {\@pglsp1@{#1}{#2}}%
    {\@pglsp1@{#1}{#2}[{}]}%
}
\def\@pglsp1@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglschasprefixplural{#2}{\glsentryprefixplural{#2}\glsprefixsep}{-}%
    }%
    {%
      \ifglschasprefixfirstplural{#2}%
        {\glsentryprefixfirstplural{#2}\glsprefixsep}{-}%
    }%
    \@glspl@{#1}{#2}[#3]%
  }%
}
}

```

```

\newrobustcmd{\Pgl}{\@gls@hyp@opt\Pgl}
\newcommand*{\@Pgl}[2][\]{%
  \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
        \@glspl@{#1}{#2}[#3]%
      }%
      {\@Glspl@{#1}{#2}[#3]}%
    }%
  }%
}

```



```

}%
{%
  \ifglshasprefixfirstplural{#2}%
  {%
    \Glsentryprefixfirstplural{#2}%
    \glsprefixsep
    \@glspl@{#1}{#2}[#3]%
  }%
  {\@Glspl@{#1}{#2}[#3]}%
}%
}%
}
\newrobustcmd{\PGLS}{\@gls@hyp@opt\PGLS}
\newcommand*{\@PGLS}[2][ ]{%
  \new@ifnextchar [%
    {\@PGLS@{#1}{#2}}%
    {\@PGLS@{#1}{#2} [ ]}%
  }
}
\def\@PGLS@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglshasprefix{#2}%
      {\mfirstucMakeUppercase{\glsentryprefix{#2}\glsprefixsep}}{%
    }%
    {%
      \ifglshasprefixfirst{#2}%
      {\mfirstucMakeUppercase{\glsentryprefixfirst{#2}\glsprefixsep}}{%
    }%
    \@GLS@{#1}{#2}[#3]%
  }%
}
}
\newrobustcmd{\PGLSpl}{\@gls@hyp@opt\PGLSpl}
\newcommand*{\@PGLSpl}[2][ ]{%
  \new@ifnextchar [%
    {\@PGLSpl@{#1}{#2}}%
    {\@PGLSpl@{#1}{#2} [ ]}%
  }
}
\def\@PGLSpl@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglshasprefixplural{#2}%
      {\mfirstucMakeUppercase{\glsentryprefixplural{#2}\glsprefixsep}}{%
    }%
    {%

```

```

        \ifglshasprefixfirstplural{#2}%
        {\mfirstucMakeUppercase{\glsentryprefixfirstplural{#2}\glsprefixsep}}{}%
    }%
    \@GLSpl@{#1}{#2}{#3}%
} %
}

```

8.8 Rollback v4.46 (glossary-hypernav-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-hypernav}[2020/03/19 v4.46 (NLCT)]
\newcommand*{\glsnavhyperlink}[3][\@glo@type]{%
  \edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%
  \glslink{\glsnavhyperlinkname{#1}{#2}}{#3}}
\newcommand*{\glsnavhyperlinkname}[2]{\glsn:#1@#2}
\newcommand*{\glsnavhypertarget}[3][\@glo@type]{%
  \glsnavhypertarget{#1}{#2}{#3}%
}
\newcommand*{\@glsnavhypertarget}[3]{%
  \protected@write\@auxout{}{\string\@gls@hypergroup{#1}{#2}}%
  \gls@target{\glsnavhyperlinkname{#1}{#2}}{#3}%
  \expandafter\let
    \expandafter\@gls@list\csname @gls@hypergroup@list@#1\endcsname
  \@for\@gls@elem:=\@gls@list\do{%
    \ifthenelse{\equal{\@gls@elem}{#2}}{\@endfortrue}{}%
  }
  \if@endfor
  \else
    \GlossariesWarningNoLine{Navigation panel
      for glossary type ‘#1’^^Jmissing group ‘#2’}%
    \gdef\gls@hypergroup@prerun{%
      \GlossariesWarningNoLine{Navigation panel
        has changed. Rerun LaTeX}}%
  \fi
}
\let\gls@hypergroup@prerun\relax
\AtEndDocument{\gls@hypergroup@prerun}
\newcommand*{\@gls@hypergroup}[2]{%
  \ifundefined{@gls@hypergroup@list@#1}{%
    \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{#2}%
  }{%
    \expandafter\let\expandafter\@gls@tmp
      \csname @gls@hypergroup@list@#1\endcsname
    \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{%
      \@gls@tmp,#2}%
  }%
}
\newcommand*{\glsnavigation}{%
  \def\@gls@between{}%
  \ifcsundef{@gls@hypergroup@list@\@glo@type}%

```

```

{%
  \def\@gls@list{%
}%
{%
  \expandafter\let\expandafter\@gls@list
  \csname @gls@hypergroup\list@\@glo@type\endcsname
}%
\@for\@gls@tmp:=\@gls@list\do{%
  \@gls@between
  \@gls@getgrouptitle{\@gls@tmp}{\@gls@grptitle}%
  \glsnavhyperlink{\@gls@tmp}{\@gls@grptitle}%
  \let\@gls@between\glshypernavsep
}%
}
\newcommand*\glshypernavsep{\space\textbar\space}
\newcommand*\glsymbolnav{%
  \glsnavhyperlink{glsymbols}{\glsgetgrouptitle{glsymbols}}%
  \glshypernavsep
  \glsnavhyperlink{glsnumbers}{\glsgetgrouptitle{glsnumbers}}%
  \glshypernavsep
}

```

8.9 Rollback v4.46 (glossary-inline-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-inline}[2020/03/19 v4.46 (NLCT)]
\newglossarystyle{inline}{%
  \renewenvironment{theglossary}%
  {%
    \def\gls@inlinesep{}%
    \def\gls@inlinesubsep{}%
    \def\gls@inlinepostchild{}%
  }%
  {\glspostinline}%
\renewcommand*\glossaryheader{}%
\renewcommand*\glsgroupheading}[1]{%
\renewcommand*\glossentry}[2]{%
  \glsinlinedopostchild
  \gls@inlinesep
  \glsentryitem{##1}%
  \glsinlinenameformat{##1}%
  \glossentryname{##1}%
}%
\ifglsdescsuppressed{##1}%
{%
  \glsinlineemptydescformat
  {%
    \glossentrysymbol{##1}%
  }%
}

```

```

    {%
      ##2%
    }%
  }%
  {%
    \ifglshasdesc{##1}%
    {\glsinlinedescformat{\glossentrydesc{##1}}{\glossentrysymbol{##1}}{##2}}%
    {\glsinlineemptydescformat{\glossentrysymbol{##1}}{##2}}%
  }%
  \ifglshaschildren{##1}%
  {%
    \glsresetsubentrycounter
    \glsinlineparentchildseparator
    \def\gls@inlinesubsep{}%
    \def\gls@inlinepostchild{\glsinlinepostchild}%
  }%
  {}%
  \def\gls@inlinesep{\glsinlineseparator}%
}%
\renewcommand{\subglossentry}[3]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{%
    \glossentryname{##2}}%
  \gls@inlinesubdescformat{\glossentrydesc{##2}}{\glossentrysymbol{##2}}{##3}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%
\renewcommand*{\gls@groupskip}{}%
}
\newcommand*{\glsinlinedopostchild}{%
  \gls@inlinepostchild
  \def\gls@inlinepostchild{}%
}
\newcommand*{\glsinlineseparator}{;\space}
\newcommand*{\glsinlinesubseparator}{,\space}
\newcommand*{\glsinlineparentchildseparator}{:\space}
\newcommand*{\glsinlinepostchild}{}
\newcommand*{\gls@postinline}{\gls@postdescription\space}
\newcommand*{\glsinlinenameformat}[2]{\glstarget{#1}{#2}}
\newcommand*{\glsinlinedescformat}[3]{\space#1}
\newcommand*{\glsinlineemptydescformat}[2]{}
\newcommand*{\glsinlinesubnameformat}[2]{\glstarget{#1}{}}
\newcommand*{\glsinlinesubdescformat}[3]{#1}

```

8.10 Rollback v4.46 (glossary-list-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-list}[2020/03/19 v4.46 (NLCT)]
\providecommand{\indexspace}{%

```

```

\par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*\glslistgroupheaderfmt}[1]{#1}
\newcommand*\glslistnavigationitem}[1]{\item[#1]}
\newglossarystyle{list}{%
  \renewenvironment{theglossary}%
    {\begin{description}}{\end{description}}%
  \renewcommand*\glossaryheader{}%
  \renewcommand*\glsgroupheading}[1]{}%
  \renewcommand*\glossentry}[2]{%
    \item[\glsentryitem{##1}]%
      \glstarget{##1}{\glossentryname{##1}}%
      \glossentrydesc{##1}\glspostdescription\space ##2}%
  \renewcommand*\subglossentry}[3]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\space
    \glossentrydesc{##2}\glspostdescription\space ##3.}%
  \renewcommand*\glsgroupskip{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{listgroup}{%
  \setglossarystyle{list}%
  \renewcommand*\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
\newglossarystyle{listhypergroup}{%
  \setglossarystyle{list}%
  \renewcommand*\glossaryheader{%
    \glslistnavigationitem{\glsnavigation}}%
  \renewcommand*\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}]}]}
\newglossarystyle{altlist}{%
  \setglossarystyle{list}%
  \renewcommand*\glossentry}[2]{%
    \item[\glsentryitem{##1}]%
      \glstarget{##1}{\glossentryname{##1}}%
      \mbox{}\par\nobreak\@afterheading
      \glossentrydesc{##1}\glspostdescription\space ##2}%
  \renewcommand*\subglossentry}[3]{%
    \par
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space ##3}%
}
\newglossarystyle{altlistgroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
\newglossarystyle{altlisthypergroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*\glossaryheader{%
    \glslistnavigationitem{\glsnavigation}}%
}

```

```

\renewcommand*{\glsgroupheading}[1]{%
  \item[\glslistgroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}]%
}
\newglossarystyle{listdotted}{%
  \setglossarystyle{list}%
  \renewcommand*{\glossentry}[2]{%
    \item[]\makebox[\glslistdottedwidth][1]{%
      \glstarget{##1}{\glossentryname{##1}}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut\glossentrydesc{##1}}%
    }
  \renewcommand*{\subglossentry}[3]{%
    \item[]\makebox[\glslistdottedwidth][1]{%
      \glssubentryitem{##2}%
      \glstarget{##2}{\glossentryname{##2}}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut\glossentrydesc{##2}}%
    }
  }
\newlength\glslistdottedwidth
\setlength{\glslistdottedwidth}{.5\hspace}
\newglossarystyle{sublistdotted}{%
  \setglossarystyle{listdotted}%
  \renewcommand*{\glossentry}[2]{%
    \item[\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}}]}%
  }
}

```

8.11 Rollback v4.46 (glossary-list-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-long}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
  \newlength\glsdescwidth
  \setlength{\glsdescwidth}{0.6\hspace}
}{%
}
\@ifundefined{glspagelistwidth}{%
  \newlength\glspagelistwidth
  \setlength{\glspagelistwidth}{0.1\hspace}
}{%
}
\newglossarystyle{long}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}[lp{\glsdescwidth}]}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand*{\glossentry}[2]{%
    \glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand*{\subglossentry}[3]{%
    &
  }
}

```

```

        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
        ##3\tabularnewline
    }%
    \ifglsnogroupskip
        \renewcommand*\{glsgroupskip}{}%
    \else
        \renewcommand*\{glsgroupskip}{ & \tabularnewline}%
    \fi
}
\newglossarystyle{longborder}{%
    \setglossarystyle{long}%
    \renewenvironment{theglossary}{%
        \begin{longtable}{|l|p{\glsdescwidth}|}{\end{longtable}}%
        \renewcommand*\{glossaryheader}{\hline\endhead\hline\endfoot}%
    }
    \newglossarystyle{longheader}{%
        \setglossarystyle{long}%
        \renewcommand*\{glossaryheader}{%
            \bfseries \entryname & \bfseries \descriptionname\tabularnewline\endhead}%
        }
    \newglossarystyle{longheaderborder}{%
        \setglossarystyle{longborder}%
        \renewcommand*\{glossaryheader}{%
            \hline\bfseries \entryname & \bfseries
            \descriptionname\tabularnewline\hline
            \endhead
            \hline\endfoot}%
        }
    \newglossarystyle{long3col}{%
        \renewenvironment{theglossary}{%
            {\begin{longtable}{|lp{\glsdescwidth}p{\glspagelistwidth}|}{\end{longtable}}%
            {\end{longtable}}%
        }%
        \renewcommand*\{glossaryheader}{}%
        \renewcommand*\{glsgroupheading}[1]{}%
        \renewcommand*\{glossentry}[2]{%
            \glssubentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
            \glossentrydesc{##1} & ##2\tabularnewline
        }%
        \renewcommand*\{subglossentry}[3]{%
            &
            \glssubentryitem{##2}%
            \glstarget{##2}{\strut}\glossentrydesc{##2} &
            ##3\tabularnewline
        }%
    \ifglsnogroupskip
        \renewcommand*\{glsgroupskip}{}%
    \else
        \renewcommand*\{glsgroupskip}{ & & \tabularnewline}%
    \fi
}

```

```

}
\newglossarystyle{long3colborder}{%
  \setglossarystyle{long3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long3colheader}{%
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long3colheaderborder}{%
  \setglossarystyle{long3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}
}
\newglossarystyle{long4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l|l|l|l}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{long4colheader}{%
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&

```



```

        \bfseries \symbolname&
        \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long4colborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long4colheaderborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}
\newglossarystyle{altlong4col}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheader}{%
  \setglossarystyle{long4colheader}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colborder}{%
  \setglossarystyle{long4colborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheaderborder}{%
  \setglossarystyle{long4colheaderborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%
    {\end{longtable}}%
}
}

```

8.12 Rollback v4.46 (glossary-longbooktabs-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-longbooktabs}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{booktabs}
\RequirePackage{glossary-long}[=v4.46]
\RequirePackage{glossary-longragged}[=v4.46]
\newglossarystyle{long-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname & \bfseries
    \descriptionname\tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{long3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname &
    \bfseries \descriptionname &
    \bfseries \pagelistname
    \tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{long4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname &
    \bfseries \descriptionname &
    \bfseries \symbolname &
    \bfseries \pagelistname
    \tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{altlong4col-booktabs}{%
  \glspatchLToutput

```

```

\setglossarystyle{long4col-booktabs}%
\renewenvironment{theglossary}%
  {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
  {\end{longtable}}%
}
\newglossarystyle{longragged-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{longragged3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlongragged4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{altlong4col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newcommand*{\glsLTpenaltycheck}{%
  \ifnum\outputpenalty=-50\vskip-\normalbaselineskip\relax\fi
}
\newcommand{\glspenaltygroupskip}{%
  \noalign{\penalty-50\vskip\normalbaselineskip}}
\let\@gls@org@LT@output\LT@output
\newcommand*{\glsrestoreLToutput}{\let\LT@output\@gls@org@LT@output}
\newcommand*{\glspatchLToutput}{%
  \renewcommand*{\LT@output}{%
    \ifnum\outputpenalty <-\@Mi
      \ifnum\outputpenalty > -\LT@end@pen
        \LT@err{floats and marginpars not allowed in a longtable}\@ehc
      \else
        \setbox\z@\vbox{\unvbox\@cclv}%
        \ifdim \ht\LT@lastfoot>\ht\LT@foot
          \dimen@pagegoal
          \advance\dimen@-\ht\LT@lastfoot
          \ifdim\dimen@<\ht\z@
            \setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%
            \@makecol
            \@outputpage
            \setbox\z@\vbox{\box\LT@head\glsLTpenaltycheck}%
          \fi
        \fi
      \fi
    }%
}

```

```

        \fi
        \fi
        \global\@colroom\@colht
        \global\vsizel\@colht
        {\unvbox\z@\box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%
        \fi
    \else
        \setbox\@cclv\vbox{\unvbox\@cclv\copy\LT@foot\vss}%
        \@makecol
        \@outputpage
        \global\vsizel\@colroom
        \copy\LT@head
        \glsLTpenaltycheck
        \nobreak
    \fi
}
}
}

```

8.13 Rollback v4.46 (glossary-longragged-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-longragged}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{array}
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
    \newlength\glsdescwidth
    \setlength\glsdescwidth{0.6\hsize}
}{%
\@ifundefined{glspagelistwidth}{%
    \newlength\glspagelistwidth
    \setlength\glspagelistwidth{0.1\hsize}
}{%
\newglossarystyle{longragged}{%
    \renewenvironment{theglossary}{%
        {\begin{longtable}{1>\raggedright}p{glsdescwidth}}}%
        {\end{longtable}}}%
    \renewcommand*\glossaryheader{}%
    \renewcommand*\glsgroupheading}[1]{}%
    \renewcommand\glossentry}[2]{%
        \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1}\glspostdescription\space ##2%
        \tabularnewline
    }%
    \renewcommand\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2}%
        \glspostdescription\space ##3%
        \tabularnewline
    }%
}
}
}

```

```

}%
\ifglsgroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip{ & \tabularnewline}%
\fi
}
\newglossarystyle{longraggedborder}{%
  \setglossarystyle{longragged}%
  \renewenvironment{theglossary}{%
    \begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}||}%
    {\end{longtable}}%
    \renewcommand*\glossaryheader{\hline\endhead\hline\endfoot}%
  }
\newglossarystyle{longraggedheader}{%
  \setglossarystyle{longragged}%
  \renewcommand*\glossaryheader{%
    \bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\endhead}%
  }
\newglossarystyle{longraggedheaderborder}{%
  \setglossarystyle{longraggedborder}%
  \renewcommand*\glossaryheader{%
    \hline\bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\hline
    \endhead
    \hline\endfoot}%
  }
\newglossarystyle{longragged3col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}%
    {\end{longtable}}%
    \renewcommand*\glossaryheader{}%
    \renewcommand*\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
      \glstarget{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1} & ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
      &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2} &
      ##3\tabularnewline
    }%
  \ifglsgroupskip
    \renewcommand*\glsgroupskip{}%
  \else
    \renewcommand*\glsgroupskip{ & & \tabularnewline}%
  \fi

```

```

}
\newglossarystyle{longragged3colborder}{%
  \setglossarystyle{longragged3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{longragged3colheader}{%
  \setglossarystyle{longragged3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{longragged3colheaderborder}{%
  \setglossarystyle{longragged3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}
\newglossarystyle{altlongragged4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{altlongragged4colheader}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%

```

```

        {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
          >{\raggedright}p{\glspagelistwidth}}}%
        {\end{longtable}}%
\renewcommand*{\glossaryheader}{%
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{altlongragged4colborder}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|l%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{altlongragged4colheaderborder}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|l%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}

```

8.14 Rollback v4.46 (glossary-mcols-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-mcols}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{multicol}
\RequirePackage{glossary-tree}[=v4.46]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\gls mcols}{2}
\newglossarystyle{mcolindex}{%
  \setglossarystyle{index}%
  \renewenvironment{theglossary}%
    {%
      \begin{multicols}{\gls mcols}
      \setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%
      \let\item\glstreeitem
      \let\subitem\glstreesubitem
      \let\subsubitem\glstreesubsubitem
    }
  }

```

```

    }%
    {\end{multicols}}%
}
\newglossarystyle{mcolindexgroup}{%
\setglossarystyle{mcolindex}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\indexspace}%
}
\newglossarystyle{mcolindexhypergroup}{%
\setglossarystyle{mcolindex}%
\renewcommand*{\glossaryheader}{%
\item\glstreenavigationfmt{\glsnavigation}\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
\indexspace}%
}
\newglossarystyle{mcolindexspannav}{%
\setglossarystyle{index}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glscols}{\noindent\glstreenavigationfmt{\glsnavigation}}
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
\let\item\glstreeitem}%
{\end{multicols}}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
\indexspace}%
}
}
\newglossarystyle{mcoltree}{%
\setglossarystyle{tree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glscols}
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%
}
\newglossarystyle{mcoltreegroup}{%
\setglossarystyle{mcoltree}%
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreehypergroup}{%
\setglossarystyle{mcoltree}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
}

```



```

\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcoltreesspannav}{%
\setglossarystyle{tree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsncols}[\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}{\glsncols}
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%
}
\newglossarystyle{mcoltreenonamegroup}{%
\setglossarystyle{mcoltreenoname}%
\renewcommand*{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreenonamehypergroup}{%
\setglossarystyle{mcoltreenoname}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcoltreenonamespannav}{%
\setglossarystyle{treenoname}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsncols}[\noindent\glstreenavigationfmt{\glsnavigation}]
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%

```

```

}%
{\end{multicols}}%
\renewcommand*{\glsgroupheading}[1]{%
  \par\noindent
  \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}
\newglossarystyle{mcolalmtree}{%
  \setglossarystyle{almtree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glscols}
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
}
\newglossarystyle{mcolalmtreegroup}{%
  \setglossarystyle{mcolalmtree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcolalmtreehypergroup}{%
  \setglossarystyle{mcolalmtree}%
  \renewcommand*{\glossaryheader}{%
    \par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcolalmtreespannav}{%
  \setglossarystyle{almtree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glscols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
}

```

```

\renewcommand*{\glsgroupheading}[1]{%
  \par
  \def\@gls@prevlevel{-1}%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}

```

8.15 Rollback v4.46 (glossary-super-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-super}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
  \newlength\glsdescwidth
  \setlength{glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength\glspagelistwidth
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{super}{%
  \renewenvironment{theglossary}%
    {\tablehead{}}\tabletail{}}%
  \begin{supertabular}{lp{glsdescwidth}}%
  {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glstryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}}%

```

```

        \begin{supertabular}{|l|p{\glsdescwidth}|}%
        {\end{supertabular}}%
    }
    \newglossarystyle{superheader}{%
        \setglossarystyle{super}%
    \renewenvironment{theglossary}%
        {\tablehead{\bfseries \entryname &
            \bfseries\descriptionname\tabularnewline}%
            \tabletail{}}%
        {\begin{supertabular}{lp{\glsdescwidth}}%
            {\end{supertabular}}%
        }
    \newglossarystyle{superheaderborder}{%
        \setglossarystyle{super}%
    \renewenvironment{theglossary}%
        {\tablehead{\hline\bfseries \entryname &
            \bfseries \descriptionname\tabularnewline\hline}%
            \tabletail{\hline}}
        {\begin{supertabular}{|l|p{\glsdescwidth}|}%
            {\end{supertabular}}%
        }
    }
    \newglossarystyle{super3col}{%
    \renewenvironment{theglossary}%
        {\tablehead{} \tabletail{}}%
        {\begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}%
            {\end{supertabular}}%
        \renewcommand*\glossaryheader{}%
        \renewcommand*\glsgroupheading}[1]{}%
        \renewcommand\glossentry[2]{%
            \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
            \glossentrydesc{##1} & ##2\tabularnewline
        }%
        \renewcommand{\subglossentry}[3]{%
            &
            \glssubentryitem{##2}%
            \glstarget{##2}{\strut}\glossentrydesc{##2} &
            ##3\tabularnewline
        }%
        \ifglsnogroupskip
            \renewcommand*\glsgroupskip}{}%
        \else
            \renewcommand*\glsgroupskip}{& & \tabularnewline}%
        \fi
    }
    \newglossarystyle{super3colborder}{%
        \setglossarystyle{super3col}%
    \renewenvironment{theglossary}%
        {\tablehead{\hline}\tabletail{\hline}%
            \begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
            {\end{supertabular}}%
        }

```

```

}
\newglossarystyle{super3colheader}{%
  \setglossarystyle{super3col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    {\begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{super3colheaderborder}{%
  \setglossarystyle{super3colborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname&\bfseries\descriptionname&
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    {\begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
    {\end{supertabular}}}%
}
\newglossarystyle{super4col}{%
  \renewenvironment{theglossary}%
    {\tablehead{}}\tabletail{}}%
    {\begin{supertabular}{llll}}{%
    \end{supertabular}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
  \fi
}
\newglossarystyle{super4colheader}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}%
    \tabletail{}}%

```

```

        \begin{supertabular}{|l|l|l|l|}%
        \end{supertabular}}%
}
\newglossarystyle{super4colborder}{%
\setglossarystyle{super4col}%
\renewenvironment{theglossary}%
{\tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}{|l|l|l|l|l|}%
\end{supertabular}}%
}
\newglossarystyle{super4colheaderborder}{%
\setglossarystyle{super4col}%
\renewenvironment{theglossary}%
{\tablehead{\hline\bfseries\tablehead{\bfseries\entryname&\bfseries\descriptionname&
\bfseries\symbolname &
\bfseries\pagelistname\tablearnewline\hline}%
\tabletail{\hline}%
\begin{supertabular}{|l|l|l|l|l|}%
\end{supertabular}}%
}
\newglossarystyle{altsuper4col}{%
\setglossarystyle{super4col}%
\renewenvironment{theglossary}%
{\tablehead{} \tabletail{}}%
\begin{supertabular}{|lp{\glsdescwidth}lp{\glspagelistwidth}}%
\end{supertabular}}%
}
\newglossarystyle{altsuper4colheader}{%
\setglossarystyle{super4colheader}%
\renewenvironment{theglossary}%
{\tablehead{\bfseries\tablehead{\bfseries\entryname&\bfseries\descriptionname&
\bfseries\symbolname &
\bfseries\pagelistname\tablearnewline}\tabletail{}}%
\begin{supertabular}{|lp{\glsdescwidth}lp{\glspagelistwidth}}%
\end{supertabular}}%
}
\newglossarystyle{altsuper4colborder}{%
\setglossarystyle{super4colborder}%
\renewenvironment{theglossary}%
{\tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}%
{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}%
\end{supertabular}}%
}
\newglossarystyle{altsuper4colheaderborder}{%
\setglossarystyle{super4colheaderborder}%
\renewenvironment{theglossary}%
{\tablehead{\hline
\bfseries\entryname &
\bfseries\descriptionname &

```

```

        \bfseries\symbolname &
        \bfseries\pagelistname\tabularnewline\hline}%
\tabletail{\hline}%
\begin{supertabular}%
  {1|p{\glsdescwidth}|1|p{\glspagelistwidth}|}%
\end{supertabular}}%
}

```

8.16 Rollback v4.46 (glossary-superragged-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-superragged}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{array}
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
  \newlength\glsdescwidth
  \setlength{glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength\glspagelistwidth
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{superragged}{%
  \renewenvironment{theglossary}%
    {\tablehead{\tabletail}}%
    {\begin{supertabular}{1>{\raggedright}p{glsdescwidth}}}%
    {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2%
    \tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3%
    \tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superraggedborder}{%
  \setglossarystyle{superragged}%
}

```

```

\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
   \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|}}%
  {\end{supertabular}}%
}
\newglossarystyle{superraggedheader}{%
  \setglossarystyle{superragged}%
\renewenvironment{theglossary}%
  {\tablehead{\bfseries \entryname & \bfseries \descriptionname
   \tabularnewline}%
   \tabletail{}}%
  \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}}}%
  {\end{supertabular}}%
}
\newglossarystyle{superraggedheaderborder}{%
  \setglossarystyle{superragged}%
\renewenvironment{theglossary}%
  {\tablehead{\hline\bfseries \entryname &
   \bfseries \descriptionname\tabularnewline\hline}%
   \tabletail{\hline}%
   \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|}}%
  {\end{supertabular}}%
}
\newglossarystyle{superragged3col}{%
\renewenvironment{theglossary}%
  {\tablehead{} \tabletail{}%
   \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}%
    >{\raggedright}p{\glspagelistwidth}}}%
  {\end{supertabular}}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  ##2\tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  ##3\tabularnewline
}%
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& \tabularnewline}%
\fi
}
\newglossarystyle{superragged3colborder}{%
  \setglossarystyle{superragged3col}%
}

```



```

\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
   \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
    >{\raggedright}p{\glspagelistwidth}|}}%
  {\end{supertabular}}%
}
\newglossarystyle{superragged3colheader}{%
  \setglossarystyle{superragged3col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
     \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    \begin{supertabular}{|l>{\raggedright}p{\glsdescwidth}%
     >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{supertabular}}%
}
\newglossarystyle{superragged3colheaderborder}{%
  \setglossarystyle{superragged3colborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
     \bfseries\entryname&\bfseries\descriptionname&
     \bfseries\pagelistname\tabularnewline\hline}%
     \tabletail{\hline}%
     \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
      >{\raggedright}p{\glspagelistwidth}|}}%
     {\end{supertabular}}%
}
\newglossarystyle{altsuperragged4col}{%
  \renewenvironment{theglossary}%
    {\tablehead{} \tabletail{}%
     \begin{supertabular}{|l>{\raggedright}p{\glsdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}|}}%
     {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
  \fi
}

```

```

}
\newglossarystyle{altsuperragged4colheader}{%
  \setglossarystyle{altsuperragged4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    {\begin{supertabular}{l>{\raggedright}p{\glsdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuperragged4colborder}{%
  \setglossarystyle{altsuper4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}%
      {l|>{\raggedright}p{\glsdescwidth}l|}%
      >{\raggedright}p{\glspagelistwidth}l}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuperragged4colheaderborder}{%
  \setglossarystyle{altsuperragged4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname &
      \bfseries\descriptionname &
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}%
      {l|>{\raggedright}p{\glsdescwidth}l|}%
      >{\raggedright}p{\glspagelistwidth}l}}%
    {\end{supertabular}}}%
}

```

8.17 Rollback v4.46 (glossary-tree-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-tree}[2020/03/19 v4.46 (NLCT)]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glstreenamefmt}[1]{\textbf{#1}}
\newcommand*{\glstreegroupheaderfmt}[1]{\glstreenamefmt{#1}}
\newcommand*{\glstreenavigationfmt}[1]{\glstreenamefmt{#1}}
\ifdef\@idxitem
{\newcommand{\glstreeitem}{\@idxitem}}
{\newcommand{\glstreeitem}{\par\hangindent40\p@}}
\ifdef\subitem

```

```

{\let\glstreesubitem\subitem}
{\newcommand\glstreesubitem{\glstreeitem\hspace*{20\p@}}
\ifdef\subsubitem
{\let\glstreesubsubitem\subsubitem}
{\newcommand\glstreesubsubitem{\glstreeitem\hspace*{30\p@}}
\newcommand{\glstreepredesc}{\space}
\newcommand{\glstreechildpredesc}{\space}
\newglossarystyle{index}{%
\renewenvironment{theglossary}%
{\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
\let\item\glstreeitem
\let\subitem\glstreesubitem
\let\subsubitem\glstreesubsubitem
}%
{\par}%
\renewcommand*\{glossaryheader}{}%
\renewcommand*\{glsgroupheading}[1]{}%
\renewcommand*\{glossentry}[2]{%
\item\glstreeitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
\ifglshassymbol{##1}{\space\glossentrysymbol{##1}}{}}%
\glstreepredesc \glossentrydesc{##1}\glspostdescription\space ##2%
}%
\renewcommand{\subglossentry}[3]{%
\ifcase##1\relax
% level 0
\item
\or
% level 1
\subitem
\glssubentryitem{##2}%
\else
% all other levels
\subsubitem
\fi
\glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
\ifglshassymbol{##2}{\space\glossentrysymbol{##2}}{}}%
\glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3%
}%
\renewcommand*\{glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
\newglossarystyle{indexgroup}{%
\setglossarystyle{index}%
\renewcommand*\{glsgroupheading}[1]{%
\item\glstreegroupheaderfmt{\glsgrouptitle{##1}}%
\indexspace
}%
}
\newglossarystyle{indexhypergroup}{%
\setglossarystyle{index}%
\renewcommand*\{glossaryheader}{%

```

```

        \item\glstreenavigationfmt{\glsnavigation}\indexspace}%
\renewcommand*\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt
        {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
    \indexspace}%
}
\newglossarystyle{tree}{%
    \renewenvironment{theglossary}%
        {\setlength{\parindent}{0pt}%
         \setlength{\parskip}{0pt plus 0.3pt}}%
        {}%
    \renewcommand*\glossaryheader{}%
    \renewcommand*\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
        \hangindent0pt\relax
        \parindent0pt\relax
        \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
        \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
        \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
    }%
    \renewcommand{\subglossentry}[3]{%
        \hangindent##1\glstreeindent\relax
        \parindent##1\glstreeindent\relax
        \ifnum##1=1\relax
            \glssubentryitem{##2}%
        \fi
        \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
        \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
        \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3\par
    }%
    \renewcommand*\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
\newglossarystyle{treegroup}{%
    \setglossarystyle{tree}%
    \renewcommand{\glsgroupheading}[1]{\par
        \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par
        \indexspace}%
    }
\newglossarystyle{treehypergroup}{%
    \setglossarystyle{tree}%
    \renewcommand*\glossaryheader}{%
        \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
    \renewcommand*\glsgroupheading}[1]{%
        \par\noindent
        \glstreegroupheaderfmt
            {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
        \indexspace}%
    }
\newlength\glstreeindent
\setlength{\glstreeindent}{10pt}
\newglossarystyle{treenoname}{%

```

```

\renewenvironment{theglossary}%
  {\setlength{\parindent}{0pt}%
   \setlength{\parskip}{0pt plus 0.3pt}}%
  {}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
  \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
  \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
}%
\renewcommand{\subglossentry}[3]{%
  \hangindent##1\glstreeindent\relax
  \parindent##1\glstreeindent\relax
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \glstarget{##2}{\strut}%
  \glossentrydesc{##2}\glspostdescription\space##3\par
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{treenonamegroup}{%
  \setglossarystyle{treenoname}%
  \renewcommand{\glsgroupheading}[1]{\par
  \noindent\glstreegroupheaderfmt
  {\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{treenonamehypergroup}{%
  \setglossarystyle{treenoname}%
  \renewcommand*{\glossaryheader}{%
  \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
  \par\noindent
  \glstreegroupheaderfmt
  {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}
\newrobustcmd*{\glsfindwidesttoplevelname}[1][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
  \forglentries[\@gls@type]{\@glo@label}%
  {%
  \ifglshasparent{\@glo@label}%
  {}%
  {%

```



```

\addtolength\glstreeindent\parindent
\parindent\glstreeindent
\else
\@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
\settowidth{\glstreeindent}{\glstreenamefmt{%
\@glswidestname\space}}{%
\settowidth{\glstreeindent}{\glstreenamefmt{%
\csname @glswidestname\romannumeral\@gls@prevlevel
\endcsname\space}}}%
\addtolength\parindent{-\glstreeindent}%
\setlength\glstreeindent\parindent
\fi
\fi
\hangindent\glstreeindent
\makebox[Opt][r]{\glstreenamebox{\gls@tmplen}{%
\glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
\ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
\glossentrydesc{##2}\glspostdescription\space ##3\par
\def\@gls@prevlevel{##1}%
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{almtreegroup}{%
\setglossarystyle{almtree}%
\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
\par\indexspace}%
}
\newglossarystyle{almtreehypergroup}{%
\setglossarystyle{almtree}%
\renewcommand*{\glossaryheader}{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}}

```

8.18 Rollback v4.49 (glossaries-2021-11-01.sty)

```
\NeedsTeXFormat{LaTeX2e}
\DeclareRelease{v4.46}{2020-03-19}{glossaries-2020-03-19.sty}
\DeclareCurrentRelease{v4.49}{2021-11-01}
\ProvidesPackage{glossaries}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{ifthen}
\RequirePackage{xkeyval}[2006/11/18]
\RequirePackage{mfirstuc}
\RequirePackage{textcase}
\renewcommand*{\mfirstucMakeUppercase}{\MakeTextUppercase}%
\RequirePackage{xfor}
\RequirePackage{datatool-base}
\RequirePackage{amsgen}
\RequirePackage{etoolbox}
\newif\if@gls@docloaded
\@ifpackageloaded{doc}{%
  {%
    \@gls@docloadedtrue
  }%
  {%
    \@ifclassloaded{nlctdoc}{\@gls@docloadedtrue}{\@gls@docloadedfalse}%
  }
}
\if@gls@docloaded
  \let\glsorg@theglossary\theglossary
  \let\glsorg@endtheglossary\endtheglossary
  \let\glsorg@PrintChanges\PrintChanges
  \renewcommand{\PrintChanges}{%
    \begingroup
      \let\theglossary\glsorg@theglossary
      \let\endtheglossary\glsorg@endtheglossary
      \glsorg@PrintChanges
    \endgroup
  }
\fi
\newif\if@gls@debug
\define@choicekey{glossaries.sty}{debug}[\gls@debug@val\gls@debug@nr]{%
  {true,false,showtargets,showaccsupp}[true]{%
    \ifcase\gls@debug@nr\relax
      % debug=true
      \@gls@debugtrue
      \renewcommand*{\GlossariesWarning}[1]{%
        \PackageWarning{glossaries}{##1}%
      }%
      \renewcommand*{\GlossariesWarningNoLine}[1]{%
        \PackageWarningNoLine{glossaries}{##1}%
      }%
      \let\gls@showtarget\@gobble
      \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
    \or
```



```

% debug=false
\@gls@debugfalse
\let\@glsshowtarget\@gobble
\let\@glsshowaccsupp\@gobblethree
\PackageInfo{glossaries}{debug mode OFF}%
\or
% debug=showtargets
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowtarget}{\@glsshowtarget}%
\or
% debug=showaccsupp
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
\fi
}
\newcommand*\glsshowtarget}[1]{%
\ifmmode
  \nfss@text{\glsshowtargetfont [#1]}%
\else
  \ifinner
    {\glsshowtargetfont [#1]}%
  \else
    \glsshowtargetouter{#1}%
  \fi
\fi
}
\newcommand*\glsshowtargetouter}[1]{%
  \glsshowtargetsymbol\marginpar{\glsshowtargetsymbol\glsshowtargetfont #1}}
\newcommand*\glsshowtargetsymbol}{\tiny$\triangleright$}
\newcommand*\glsshowtargetfont}{\ttfamily\footnotesize}
\newcommand*\@glsshowtarget}[1]{%
\newrobustcmd*\@@glsshowtarget}[1]{%
  \begingroup
  \protected@edef\@gls@tmp{#1}%
  \@onelevel@sanitize\@gls@tmp
  \expandafter\glsshowtarget\expandafter{\@gls@tmp}%

```

```

\endgroup
}
\newcommand*{\@glsshowaccsupp}[3]{}
\newrobustcmd*{\glsshowaccsupp}[3]{%
  \ifstrempy{#1}%
  {\@glsshowtarget{/#2 (#3)}}%
  {\@glsshowtarget{/#2 (#3) [#1]}}%
}
\newcommand*{\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries\space (or move
  \string\newglossaryentry\space
  definitions into the preamble)}%
  {You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key. It may
  be that the 'see' key has been written to the .glsdefs
  file from the previous run, in which case you need to
  move your definitions
  to the preamble if you don't want to use
  \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoinde}x{%
  [\gls@seenoinde@val\gls@seenoinde@nr]{error,warn,ignore}{%
    \ifcase\gls@seenoinde@nr
      \renewcommand*{\@gls@see@noindex}{%
        \PackageError{glossaries}%
        {'\gls@xr@key' key may only be used after \string\makeglossaries\space
        or \string\makenoidxglossaries}%
        {You must use \string\makeglossaries\space
        or \string\makenoidxglossaries\space before defining
        any entries that have a '\gls@xr@key' key}%
      }%
    \or
      \renewcommand*{\@gls@see@noindex}{%
        \GlossariesWarning{'\gls@xr@key' key ignored}%
      }%
    \or
      \renewcommand*{\@gls@see@noindex}{}%
    \fi
  }
\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}{
  {\newcommand*{\@@glossarysec}{section}}%
  {\newcommand*{\@@glossarysec}{chapter}}
}
\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
subsection,subsubsection,paragraph,subparagraph}[section]{%

```

```

\renewcommand*{\@glossarysec}{#1}}
\newcommand*{\@glossarysecstar}{*}
\newcommand*{\@glossaryseclabel}{}
\newcommand*{\glsautoprefix}{}
\define@choicekey{glossaries.sty}{numberedsection}%
  [\gls@numberedsection@val\gls@numberedsection@nr]{%
false,nolabel,autolabel,nameref}[nolabel]{%
  \ifcase\gls@numberedsection@nr\relax
    \renewcommand*{\@glossarysecstar}{*}%
    \renewcommand*{\@glossaryseclabel}{}%
  \or
    \renewcommand*{\@glossarysecstar}{}%
    \renewcommand*{\@glossaryseclabel}{}%
  \or
    \renewcommand*{\@glossarysecstar}{}%
    \renewcommand*{\@glossaryseclabel}{}%
    \label{\glsautoprefix\@glo@type}}%
  \or
    \renewcommand*{\@glossarysecstar}{*}%
    \renewcommand*{\@glossaryseclabel}{}%
    \protected@edef\@currentlabelname{\glossarytoctitle}%
    \label{\glsautoprefix\@glo@type}}%
  \fi
}
\@ifpackageloaded{classicthesis}
{\newcommand*{\@glossary@default@style}{index}}
{\newcommand*{\@glossary@default@style}{list}}
\define@key{glossaries.sty}{style}{%
  \def\@glossary@default@style{#1}%
}
\newcommand*{\@gls@declareoption}[2]{%
  \DeclareOptionX{#1}{#2}%
  \DeclareOption{#1}{#2}%
}
\newcommand*{\glossaryentrynumbers}[1]{#1\gls@save@numberlist{#1}}
\@gls@declareoption{nonumberlist}{%
  \renewcommand*{\glossaryentrynumbers}[1]{\gls@save@numberlist{#1}}%
}
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{}
\gls@savenumberlistfalse
\newcommand*\@glo@seeautonumberlist{}
\@gls@declareoption{seeautonumberlist}{%
  \renewcommand*\@glo@seeautonumberlist{}%
  \def\@glo@prefix{\glsnextpages}%
}%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{}
\gls@esclocationstrue
\newcommand*{\@gls@loadlong}{\RequirePackage{glossary-long}[=v4.49]}
\@gls@declareoption{no long}{\renewcommand*{\@gls@loadlong}{}

```

```

\IfFileExists{supertabular.sty}{%
  \newcommand*{\@gls@loadsuper}{\RequirePackage{glossary-super}[=v4.49]}{%
  \newcommand*{\@gls@loadsuper}{}}
\@gls@declareoption{nosuper}{\renewcommand*{\@gls@loadsuper}{}}
\newcommand*{\@gls@loadlist}{\RequirePackage{glossary-list}[=v4.49]}
\@gls@declareoption{no list}{%
  \renewcommand*{\@gls@loadlist}{%
  \ifdefstring{\@glossary@default@style}{list}%
  {\let\@glossary@default@style\relax}%
  }%
}%
}
\newcommand*{\@gls@loadtree}{\RequirePackage{glossary-tree}[=v4.49]}
\@gls@declareoption{notree}{\renewcommand*{\@gls@loadtree}{}}
\@gls@declareoption{nostyles}{%
  \renewcommand*{\@gls@loadlong}{}%
  \renewcommand*{\@gls@loadsuper}{}%
  \renewcommand*{\@gls@loadlist}{}%
  \renewcommand*{\@gls@loadtree}{}%
  \let\@glossary@default@style\relax
}
\newcommand*{\glspostdescription}{%
  \ifglsnopostdot\else.\spacefactor\sfcode‘\.\ \fi
}
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{}
\glsnopostdotfalse
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{}
\glsnogroupskipfalse
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{}
\@ifclassloaded{memoir}
{%
  \glsucmarktrue
}%
{%
  \glsucmarkfalse
}
\newcommand*{\@gls@define@glossaryentrycounter}{%
  \ifglsentrycounter
  \ifundef\c@glossaryentry
  {%
    \ifx\@gls@counterwithin\@empty
    \newcounter{glossaryentry}%
    \else
    \newcounter{glossaryentry}[\@gls@counterwithin]%
    \fi
  \def\theHglossaryentry{\currentglossary.\theglossaryentry}%
  }%
  }%
\fi
}

```

```

\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\glentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
  \renewcommand*{\@gls@counterwithin}{#1}%
  \glentrycountertrue
  \@gls@define@glossaryentrycounter
}
\newcommand*{\@gls@counterwithin}{}
\newcommand{\@gls@define@glossarysubentrycounter}{%
  \ifundef\c@glossarysubentry
  {%
    \ifglssubentrycounter
    \ifglentrycounter
      \newcounter{glossarysubentry}[glossaryentry]%
    \else
      \newcounter{glossarysubentry}%
    \fi
    \def\theH@glossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
  \fi
  }%
  }%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
\glssubentrycounterfalse
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none}{%
  \renewcommand*{\@glo@default@sorttype}{#1}%
  \csname @gls@setupsort@#1\endcsname
}
\newcommand*{\@gls@prestandardsort}[3]{%
  \glsdosanitizesort
}
\newcommand*{\@glo@check@sortallowed}[1]{}
\newcommand*{\@gls@setupsort@standard}{%
  \def\do@glo@storeentry{\@glo@storeentry}%
  \def\@gls@defsortcount##1{%
  \def\@gls@defsort##1##2{%
    \ifx\@glo@sort\@glsdefaultsort
      \let\@glo@sort\@glo@name
    \fi
    \let\glsdosanitizesort\@gls@sanitizesort
    \glsprestandardsort{\@glo@sort}{##1}{##2}%
    \expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
  }%
  \def\@gls@setsort##1{%
  \let\@glo@check@sortallowed\@gobble
}
\@gls@setupsort@standard
\newcommand*\glsortnumberfmt[1]{%
  \ifnum#1<100000 0\fi

```

```

\ifnum#1<10000 0\fi
\ifnum#1<1000 0\fi
\ifnum#1<100 0\fi
\ifnum#1<10 0\fi
\number#1%
}
\newcommand*{\@gls@setupsort@def}{%
\def\do@glo@storeentry{\@glo@storeentry}%
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\ifcsundef{glossary@##1@sortcount}%
{\@gls@defsortcount{##1}}%
{}%
\expandafter\global\expandafter
\advance\csname glossary@##1@sortcount\endcsname by 1\relax
\expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
\expandafter\glssortnumberfmt
{\csname glossary@##1@sortcount\endcsname}}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*{\@gls@setupsort@use}{%
\let\do@glo@storeentry\@gobble
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\expandafter\gdef\csname glo@##2@sort\endcsname{}%
}%
\def\@gls@setsort##1{%
\protected@edef\@glo@parent{\csname glo@##1@parent\endcsname}%
\ifx\@glo@parent\@empty
\else
\expandafter\@gls@setsort\expandafter{\@glo@parent}%
\fi
\protected@edef\@glo@type{\csname glo@##1@type\endcsname}%
\protected@edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
\ifx\@gls@tmp\@empty
\expandafter\global\expandafter
\advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
\expandafter\protected@xdef\csname glo@##1@sort\endcsname{%
\expandafter\glssortnumberfmt
{\csname glossary@\@glo@type @sortcount\endcsname}}%
\@glo@storeentry{##1}%
\fi

```

```

}%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*\@gls@setupsort@none{%
\def\do@glo@storeentry##1{%
\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
\expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort
}%
\def\@gls@setsort##1{%
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
{Option sort=none not allowed with \string##1}%
{(Use sort=def instead)}}%
}
\newcommand*\glsdefmain{%
\if@gls@docloaded
\newglossary[glg2]{main}{gls2}{glo2}{\glossaryname}%
\else
\newglossary{main}{gls}{glo}{\glossaryname}%
\fi
\newcommand*\gls@tr@set@main@toctitle{%
\translatelet{\glossarytoctitle}{Glossary}%
}%
}
\newcommand*\glsdefaulttype{main}
\newcommand*\acronymtype{\glsdefaulttype}
\@gls@declareoption{nomain}{%
\let\glsdefaulttype\relax
\renewcommand*\glsdefmain}{%
}
\define@boolkey{glossaries.sty}[gls]{acronym}[true]{%
\ifglsacronym
\renewcommand*\@gls@do@acronymsdef{%
\DeclareAcronymList{acronym}%
\newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
\renewcommand*\acronymtype{acronym}%
\newcommand*\gls@tr@set@acronym@toctitle{%
\translatelet{\glossarytoctitle}{Acronyms}%
}%
}%
\else
\let\@gls@do@acronymsdef\relax
\fi
}
\AtBeginDocument{%
\ifglsacronym
\ifbool{glscompatible-3.07}%
{}%
{%
\providecommand*\printacronyms[1]{}%
}
}

```

```

        \printglossary[type=\acronymtype,#1]}%
    }%
\fi
}
\newcommand*{\@gls@do@acronymsdef}{}
\@gls@declareoption{acronyms}{%
  \glsacronymtrue
  \def\@gls@do@acronymsdef{%
    \DeclareAcronymList{acronym}%
    \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
    \renewcommand*{\acronymtype}{acronym}%
    \newcommand*\gls@tr@set@acronym@toctitle}{%
      \translatelet{\glossarytoctitle}{Acronyms}%
    }%
  }%
}
\newcommand*{\@gls@acronymlists}{}
\newcommand*{\@addtoacronymlists}[1]{%
  \ifx\@gls@acronymlists\@empty
    \protected@xdef\@gls@acronymlists{#1}%
  \else
    \protected@xdef\@gls@acronymlists{\@gls@acronymlists,#1}%
  \fi
}
\newcommand*\DeclareAcronymList}[1]{%
  \glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}
\newcommand*\glsIfListOfAcronyms}[1]{%
  \protected@edef\@do@gls@islistofacronyms{%
    \noexpand\@gls@islistofacronyms{#1}{\@gls@acronymlists}}%
  \@do@gls@islistofacronyms
}
\newcommand*\@gls@islistofacronyms}[4]{%
  \def\gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%
    \def\@gls@before{##1}\def\@gls@after{##2}}%
  \gls@islistofacronyms,#2,#1,\@nil\end@gls@islistofacronyms
  \ifx\@gls@after\@nnil
    #4%
  \else
    #3%
  \fi
}
\newif\if@gls@isacronymlist
\newcommand*{\gls@checkisacronymlist}[1]{%
  \glsIfListOfAcronyms{#1}%
  {\@gls@isacronymlisttrue}{\@gls@isacronymlistfalse}%
}
\newcommand*\SetAcronymLists}[1]{%
  \renewcommand*\@gls@acronymlists{#1}%
}

```



```

\define@key{glossaries.sty}{acronymlists}{%
  \DeclareAcronymList{#1}%
}
\newcommand{\glscounter}{page}
\define@key{glossaries.sty}{counter}{%
  \renewcommand*{\glscounter}{#1}%
}
\newcommand*{\@gls@nohyperlist}{}
\newcommand*{\GlsDeclareNoHyperList}[1]{%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*{\@gls@nohyperlist}{#1}%
  }%
  {%
    \appto\@gls@nohyperlist{,#1}%
  }%
}
\define@key{glossaries.sty}{nohypertypes}{%
  \GlsDeclareNoHyperList{#1}%
}
\newcommand*{\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{#1}%
}
\newcommand*{\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{#1}%
}
\newcommand{\glosortentrieswarning}{%
  \typeout{Using TeX to sort glossary entries---this may
  take a while}%
}
\@gls@declareoption{nowarn}{%
  \if@gls@debug
    \GlossariesWarning{Warnings can't be suppressed in debug mode}%
  \else
    \renewcommand*{\GlossariesWarning}[1]{}%
    \renewcommand*{\GlossariesWarningNoLine}[1]{}%
    \renewcommand*{\glosortentrieswarning}{}%
    \renewcommand*{\@gls@missinglang@warn}[2]{}%
  \fi
}
\newcommand*{\@gls@missinglang@warn}[2]{%
  \PackageWarningNoLine{glossaries}%
  {No language module detected for '#1'.\MessageBreak
  Language modules need to be installed separately.\MessageBreak
  Please check on CTAN for a bundle called\MessageBreak
  'glossaries-#2' or similar}%
}
\@gls@declareoption{nolangwarn}{%
  \renewcommand*{\@gls@missinglang@warn}[2]{}%
}

```

```

\newcommand*{\@gls@warnonglossdefined}{%
  \GlossariesWarning{Overriding \string\printglossary}%
}
\newcommand*{\@gls@warnontheGLOSSdefined}{%
  \GlossariesWarning{Overriding 'theGLOSS' environment}%
}
\@gls@declareoption{noredefwarn}{%
  \renewcommand*{\@gls@warnonglossdefined}{}%
  \renewcommand*{\@gls@warnontheGLOSSdefined}{}%
}
\newcommand*{\@gls@sanitizedesc}{%
}
\newcommand*{\glssetexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@expand@field{##1}{#1}{##2}%
  }%
}
\newcommand*{\glssetnoexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@noexpand@field{##1}{#1}{##2}%
  }%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}
\glssetnoexpandfield{descplural}
\newcommand*{\@gls@sanitizename}{}
\glssetnoexpandfield{name}
\newcommand*{\@gls@sanitizesymbol}{}
\glssetnoexpandfield{symbol}
\glssetnoexpandfield{symbolplural}
\newcommand*{\@gls@sanitizesort}{%
  \ifgls@sanitizesort
    \@gls@sanitizesort
  \else
    \@gls@nosanitizesort
  \fi
}
\newcommand*{\@gls@sanitizesort}{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@gls@nosanitizesort}{}
\newcommand*{\@gls@noidx@sanitizesort}{%
  \ifdefvoid\@glo@sort
  {}%
  {%
    \expandafter\@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
  \def\@glo@sort{#1#2}%
}

```

```

\@onelevel@sanitize\@glo@sort
}
\newcommand*{\@@gls@noidx@nosanitizesort}{%
\ifdefvoid\@glo@sort
{}%
{}%
\expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
}%
}
\def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
\bgroup
\glsnoidxstripaccents
\protected@xdef\@@glo@sort{#1#2}%
\egroup
\let\@glo@sort\@@glo@sort
}
\newcommand*\glsnoidxstripaccents{%
\let\IeC\@firstofone
\let\add@accent@\@secondoftwo
\let\@text@composite@x\@secondoftwo
\let\@tabacckludge\@secondoftwo
\expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
\let'\@firstofone
\let'\@firstofone
\let~\@firstofone
\let"\@firstofone
\let\u\@firstofone
\let\t\@firstofone
\let\d\@firstofone
\let\r\@firstofone
\let=\@firstofone
\let.\@firstofone
\let~\@firstofone
\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{\languagename}%
\define@key[gls]{xindy}{language}{\def\@xdy@main@language{#1}}
\ifcsundef{inputencodingname}{%
\def\gls@codepage{}}{%
\def\gls@codepage{\inputencodingname}
}
\define@key[gls]{xindy}{codepage}{\def\gls@codepage{#1}}
\define@key{glossaries.sty}{xindy}[]{%
\glsxindytrue
\setkeys[gls]{xindy}{#1}%
}
\@gls@declareoption{xindygloss}{%
\glsxindytrue
}
\@gls@declareoption{xindynoglsnumbers}{%
\glsxindytrue
\gls@xindy@glsnumbersfalse
}
\providecommand{\@domakeglossaries}[1]{#1}
\@gls@declareoption{disablemakegloss}{%
\ifdefequal\makeglossaries\@no@makeglossaries
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect
(\string\makenoidxglossaries\space has already been used)}%
}%
{%
\ifdefequal\@makeglossary\@gobble
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect
(\string\makeglossaries\space has already been used)}%
}%
{%
\renewcommand{\@domakeglossaries}[1]{%
\PackageInfo{glossaries}{\string\makeglossaries\space and
\string\makenoidxglossaries\space have been disabled}%
}%
}%
}%
}
\@gls@declareoption{restoremakegloss}{%
\ifdefequal\makeglossaries\@no@makeglossaries
{%
\GlossariesWarning{Option ‘restoremakegloss’ has no effect
(\string\makenoidxglossaries\space has already been used)}%
}
}

```



```

}%
{%
  \ifdefequal\@makeglossary\@gobble
  {%
    \GlossariesWarning{Option ‘restoremakegloss’ has no effect
      (\string\makeglossaries\space has already been used)}%
  }%
  {%
    \PackageInfo{glossaries}{\string\makeglossaries\space and
      \string\makenoidxglossaries\space have been restored}%
    \let\@domakeglossaries\@firstofone
  }%
}%
}
\newcommand*\@do@write@glslabels{%
  \AtEndDocument{\@do@write@glslabels}%
  \let\@do@write@glslabels\relax
}
\newcommand*\@@do@write@glslabels{%
  \newwrite\@glsl@labelsfile
  \immediate\openout\@glsl@labelsfile=\jobname.glslabels
  \protected@edef\@glsl@fulllist{\@glo@types}%
  \ifdefempty{\@ignored@glossaries}{}%
  {\protected@eappto\@glsl@fulllist{,\@ignored@glossaries}}%
  \forallglstentries[\@glsl@fulllist]{\@glstentry}%
  {%
    \ifdefempty{\@glstentry}{}%
    {%
      \ifx\do@write@glslabelsname\@gobble
        \def\@glst@thisentryname{%
      \else
        \letcs{\@glst@thisentryname}{glo@\glstetoklabel{\@glstentry}@name}%
        \@onelevel@sanitize\@glst@thisentryname
      \fi
      \immediate\write\@glsl@labelsfile
        {\@glstentry\do@write@glslabelsname{\@glst@thisentryname}}%
    }%
  }%
  \immediate\closeout\@glsl@labelsfile
}
\newcommand*\do@write@glslabelsname}[1]{%
  \begingroup
  \catcode'\^^I12
  \gdef\@write@glslabelsname#1{\^^I#1}
  \endgroup
  \@glsl@declareoption{writeglslabels}{\@do@write@glslabels
    \let\do@write@glslabelsname\@gobble}
  \@glsl@declareoption{writeglslabelnames}{\@do@write@glslabels
    \let\do@write@glslabelsname\@write@glslabelsname}
  \newif\ifglslautomake

```

```

\newcommand{\gls@automake@nr}{1}
\define@choicekey{glossaries.sty}{automake}%
  [\gls@automake@val\gls@automake@nr]{true,false,immediate}[true]{%
  \ifnum\gls@automake@nr=1\relax
    \glsautomakefalse
  \else
    \glsautomaketrue
  \fi
  \ifglsautomake
    \renewcommand*\@gls@doautomake}{%
      \PackageError{glossaries}{You must use
      \string\makeglossaries\space with automake=true}
      {%
        Either remove the automake=true setting or
        add \string\makeglossaries\space to your document preamble.%
      }%
    }%
  \else
    \renewcommand*\@gls@doautomake}{}%
  \fi
}
\glsautomakefalse
\newcommand*\@gls@doautomake{}
\AtEndDocument{\@gls@doautomake}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
  \ifgls savewrites
    \renewcommand*\@gls@writefiles{\@gls@writefiles}%
  \else
    \let\@gls@writefiles\@empty
  \fi
}
\gls savewritesfalse
\let\@gls@writefiles\@empty
\define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{%
\boolfalse{glscompatible-3.07}
\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
  \ifbool{glscompatible-2.07}%
  {%
    \booltrue{glscompatible-3.07}%
  }%
  }%
}
\boolfalse{glscompatible-2.07}
\let\@gls@original@makeglossary\makeglossary
\let\@gls@original@glossary@glossary
\def\makeglossary{%
  \GlossariesWarning{Use of \string\makeglossary\space with
  glossaries.sty is \MessageBreak deprecated. Use \string\makeglossaries\space
  instead. If you \MessageBreak need the original definition of
  \string\makeglossary\space use \MessageBreak the package options

```

```

kernelglossredefs=false (to \MessageBreak restore the former definition of
\string\makeglossary) and \MessageBreak nomain (if the file extensions cause a
conflict))%
\makeglossaries
}
\newcommand*{\@gls@override@glossary}[1][main]{%
\GlossariesWarning{Use of \string\glossary\space with
glossaries.sty is deprecated. \MessageBreak Indexing should be performed
with the user level \MessageBreak commands, such as \string\gls\space or
\string\glsadd. If you need the \MessageBreak original definition of
\string\glossary\space use the package \MessageBreak options
kernelglossredefs=false (to restore the \MessageBreak former definition of
\string\glossary) and nomain (if the \MessageBreak file extensions cause a
conflict))%
\gls@glossary{#1}%
}
\if@gls@docloaded
\else
\def\glossary{\@gls@override@glossary}
\fi
\define@choicekey{glossaries.sty}{kernelglossredefs}%
[\gls@debug@val\gls@debug@nr]{true,false,nowarn}[true]%
{%
\ifcase\gls@debug@nr\relax
\def\glossary{\@gls@override@glossary}%
\def\makeglossary{%
\GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is deprecated. Use \string\makeglossaries\space
instead. If you need the original definition of
\string\makeglossary\space use the package options
kernelglossredefs=false (to prevent redefinition of
\string\makeglossary) and nomain (if the file extensions cause a
conflict))%
\makeglossaries
}%
\or
\let\glossary\gls@original@glossary
\let\makeglossary\gls@original@makeglossary
\or
\def\makeglossary{\makeglossaries}%
\renewcommand*{\@gls@override@glossary}[1][main]{%
\gls@glossary{##1}%
}%
\fi
}
\@gls@declareoption{symbols}{%
\let\@gls@do@symbolsdef\@gls@symbolsdef
}
\newcommand*{\@gls@do@symbolsdef}{}
\newcommand*{\@gls@symbolsdef}{%

```

```

\newglossary[slg]{symbols}{sls}{slo}{\glssymbolsgroupname}%
\newcommand*\{printsymbols}[1][\]{\printglossary[type=symbols,##1]}%
\newcommand*\{gls@tr@set@symbols@toctitle}{%
  \translatelet{\glossarytoctitle}{Symbols (glossaries)}%
}%
}%
\@gls@declareoption{numbers}{%
  \let\@gls@do@numbersdef\@gls@numbersdef
}
\newcommand*\{gls@do@numbersdef}{%
\newcommand*\{gls@numbersdef}{%
  \newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
  \newcommand*\{printnumbers}[1][\]{\printglossary[type=numbers,##1]}%
  \newcommand*\{gls@tr@set@numbers@toctitle}{%
    \translatelet{\glossarytoctitle}{Numbers (glossaries)}%
  }%
}%
\@gls@declareoption{index}{%
  \ifx\@gls@do@indexdef\@empty
    \let\@gls@do@indexdef\@gls@indexdef
  \fi
}
\@gls@declareoption{noglossaryindex}{%
  \let\@gls@do@indexdef\relax
}
\newcommand*\{gls@do@indexdef}{%
\newcommand*\{gls@indexdef}{%
  \newglossary[ilg]{index}{ind}{idx}{\indexname}%
  \newcommand*\{printindex}[1][\]{\printglossary[type=index,##1]}%
  \newcommand*\{newterm}[2][\]{%
    \newglossaryentry{##2}%
    {type={index},name={##2},description={\nopostdesc},##1}}
  \let\@gls@do@indexdef\relax
}%
\@for\CurrentOption :=\@declaredoptions\do{%
  \ifx\CurrentOption\@empty
    \else
      \@expandtwoargs
        \in@ {,\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
      \ifin@
        \@use@ption
        \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
      \fi
    \fi
}
\ProcessOptionsX
\RequirePackage{glossaries-compatible-307}[v4.49]
\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,notranslate,nolong,nosuper,notree,nostyles,%

```

```

nomain,noglossaryindex}
\newcommand*\setupglossaries[1]{%
  \renewcommand*\@gls@setacrstyle}{}%
  \ifglsacrshortcuts
    \def\@gls@setupshortcuts{\glsacrshortcutstrue}%
  \else
    \def\@gls@setupshortcuts{%
      \ifglsacrshortcuts
        \DefineAcronymSynonyms
      \fi
    }%
  \fi
  \glsacrshortcutsfalse
  \let\@gls@do@numbersdef\relax
  \let\@gls@do@symbolssdef\relax
  \let\@gls@do@indexdef\relax
  \let\@gls@do@acronymsdef\relax
  \ifglsentrycounter
    \let\@gls@doentrycounterdef\relax
  \else
    \let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
  \fi
  \ifglssubentrycounter
    \let\@gls@dosubentrycounterdef\relax
  \else
    \let\@gls@dosubentrycounterdef\@gls@define@glossarysubentrycounter
  \fi
  \setkeys{glossaries.sty}{#1}%
  \@gls@setacrstyle
  \@gls@setupshortcuts
  \@gls@do@acronymsdef
  \@gls@do@numbersdef
  \@gls@do@symbolssdef
  \@gls@do@indexdef
  \@gls@doentrycounterdef
  \@gls@dosubentrycounterdef
}
\ifthenelse{\equal{\glscounter}{section}}{%
  {%
    \ifcsundef{chapter}{}%
    {%
      \let\@gls@old@chapter\@chapter
      \def\@chapter[#1]#2{\@gls@old@chapter[#1]}{#2}%
      \ifcsundef{hyperdef}{\hyperdef{section}{\thesection}}{}}%
    }%
  }%
  {}
\newcommand*\@gls@onlypremakeg{}
\newcommand*\@onlypremakeg[1]{%
  \ifx\@gls@onlypremakeg\@empty

```

```

        \def\@gls@onlypremakeg{#1}%
    \else
        \expandafter\toks@\expandafter{\@gls@onlypremakeg}%
        \edef\@gls@onlypremakeg{\the\toks@,\noexpand#1}%
    \fi
}
\newcommand*{\@disable@onlypremakeg}{%
\@for\@thiscs:=\@gls@onlypremakeg\do{%
    \expandafter\@disable@premakecs\@thiscs%
}}
\newcommand*{\@disable@premakecs}[1]{%
    \def#1{\PackageError{glossaries}{\string#1\space may only be
    used before \string\makeglossaries}{You can't use
    \string#1\space after \string\makeglossaries}}%
}
\providecommand*\glossaryname{Glossary}
\providecommand*\acronymname{Acronyms}
\newcommand*\glssettoctitle[1]{%
    \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}
\providecommand*\entryname{Notation}
\providecommand*\descriptionname{Description}
\providecommand*\symbolname{Symbol}
\providecommand*\pagelistname{Page List}
\providecommand*\glsymbolsgroupname{Symbols}
\providecommand*\glsnumbersgroupname{Numbers}
\newcommand*\glspluralsuffix}{s}
\newcommand*\glsacrpluralsuffix}{\glspluralsuffix}
\newcommand*\glsupacrpluralsuffix}{\glstextup{\glsacrpluralsuffix}}
\providecommand*\seename}{see}
\providecommand*\andname}{\&}
\newcommand*\RequireGlossariesLang}[1]{%
    \ifundefined{ver@glossaries-#1.ldf}{\input{glossaries-#1.ldf}}{}}
}
\newcommand*\ProvidesGlossariesLang}[1]{%
    \ProvidesFile{glossaries-#1.ldf}%
}
\newcommand*\addglossarytocaptions}[1]{}
\ifglstranslate
    \RequirePackage{tracklang}
    \@gls@usetranslator
    \ifpackageloaded{translator}
    {%
        \ifboolexpr
        {
            test {\ifdefstring{\trans@languages}{English}}
            and not
            test {\ifdefstring{bbl@loaded}{english}}
        }
        {%
            \let\glsifusetranslator\@secondoftwo

```

```

}%
{%
\usedictionary{glossaries-dictionary}%
\renewcommand*{\addglossarytocaptions}[1]{%
\ifcsundef{captions#1}{}%
{%
\expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
\expandafter\toks@\expandafter{\@gls@tmp
\renewcommand*{\glossaryname}{\translate{Glossary}}}%
}%
\expandafter\edef\csname captions#1\endcsname{\the\toks@}%
}%
}%
}%
{}%
\AnyTrackedLanguages
{%
\ForEachTrackedDialect{\this@dialect}{%
\IfTrackedLanguageFileExists{\this@dialect}%
{glossaries-}% prefix
{.ldf}%
{%
\RequireGlossariesLang{\CurrentTrackedTag}%
}%
{%
\@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
}%
}%
}%
{}%
\glsifusetranslator
{%
\renewcommand*{\glssettoctitle}[1]{%
\ifcsdef{gls@tr@set@#1@toctitle}%
{%
\csuse{gls@tr@set@#1@toctitle}%
}%
{%
\def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
}%
}%
\renewcommand*{\glossaryname}{\translate{Glossary}}%
\renewcommand*{\acronymname}{\translate{Acronyms}}%
\renewcommand*{\entryname}{\translate{Notation (glossaries)}}%
\renewcommand*{\descriptionname}{%
\translate{Description (glossaries)}}%
\renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}%
\renewcommand*{\pagelistname}{%
\translate{Page List (glossaries)}}%

```

```

        \renewcommand*{\glssymbolsgroupname}{%
          \translate{Symbols (glossaries)}}%
        \renewcommand*{\glsnumbersgroupname}{%
          \translate{Numbers (glossaries)}}%
      }{}%
\fi
\DeclareRobustCommand*{\nopostdesc}{}
\newcommand*{\@nopostdesc}{%
  \let\org@glspostdescription\glspostdescription
  \def\glspostdescription{%
    \let\glspostdescription\org@glspostdescription}%
}
\newcommand*{\@no@post@desc}{\nopostdesc}
\newcommand{\glspar}{\par}
\newcommand{\setStyleFile}[1]{%
  \renewcommand*{\gls@istfilebase}{#1}%
  \ifglxindy
    \def\istfilename{\gls@istfilebase.xdy}
  \else
    \def\istfilename{\gls@istfilebase.ist}
  \fi
}
\@onlypremakeg\setStyleFile
\ifglxindy
  \def\istfilename{\gls@istfilebase.xdy}
\else
  \def\istfilename{\gls@istfilebase.ist}
\fi
\newcommand*{\gls@istfilebase}{\jobname}
\newcommand*{\@istfilename}[1]{}
\newcommand*{\glscompositor}{.}
\newcommand*{\glsSetCompositor}[1]{%
  \renewcommand*{\glscompositor}{#1}}
\@onlypremakeg\glsSetCompositor
\newcommand*{\@glsAlphaCompositor}{\glscompositor}
\ifglxindy
  \newcommand*\glsSetAlphaCompositor[1]{%
    \renewcommand*\@glsAlphaCompositor{#1}}
\else
  \newcommand*\glsSetAlphaCompositor[1]{%
    \glsnoxindywarning\glsSetAlphaCompositor}
\fi
\@onlypremakeg\glsSetAlphaCompositor
\newcommand*{\gls@suffixF}{}
\newcommand*{\glsSetSuffixF}[1]{%
  \renewcommand*{\gls@suffixF}{#1}}
\@onlypremakeg\glsSetSuffixF
\newcommand*{\gls@suffixFF}{}
\newcommand*{\glsSetSuffixFF}[1]{%
  \renewcommand*{\gls@suffixFF}{#1}}

```



```

}
\ifcsundef{hyperlink}%
{%
  \newcommand*\glsnumberformat}[1]{#1}%
}%
{%
  \newcommand*\glsnumberformat}[1]{\glsnumberformat{#1}}%
}
\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*\glossarypreamble{%
  \csuse{@glossarypreamble@\currentglossary}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaulttype]{%
  \ifglossaryexists*{#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*\glossarypostamble{}
\newcommand*\glossarysection}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@p@glossarysection{#1}{#2}%
  }%
  \gls@glossarymark{\glossarytoctitle}%
}
\ifcsundef{glossarymark}%
{%
  \newcommand*\gls@glossarymark}[1]{\glossarymark{#1}}
}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand*\gls@glossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUchead{#1}}{\memUchead{#1}}%
      \else
        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand*\gls@glossarymark}[1]{%
      \ifglsucmark
        \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
      \else

```

```

        \@mkboth{#1}{#1}%
    \fi
}
}
\providecommand{\glossarymark}[1]{%
    \ifglsucmark
        \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
    \else
        \@mkboth{#1}{#1}%
    \fi
}
\newcommand*\setglossarysection[1]{%
\setkeys{glossaries.sty}{section=#1}}
\newcommand*{@glossarysection}[2]{%
    \ifdefempty{@glossarysecstar
    {%
        \csname@@glossarysec\endcsname[#1]{#2}%
    }%
    {%
        \csname@@glossarysec\endcsname*{#2}%
        \@gls@toc{#1}{\@@glossarysec}%
    }%
    \@glossaryseclabel
}
\newcommand*{@p@glossarysection}[2]{%
    \glsclearpage
    \phantomsection
    \ifdefempty{@glossarysecstar
    {%
        \csname@@glossarysec\endcsname{#2}%
    }%
    {%
        \@gls@toc{#1}{\@@glossarysec}%
        \csname@@glossarysec\endcsname*{#2}%
    }%
    \@glossaryseclabel
}
\newcommand*\gls@docclearpage{%
    \ifthenelse{\equal{\@@glossarysec}{chapter}}%
    {%
        \ifcsundef{cleardoublepage}%
        {%
            \clearpage
        }%
        {%
            \ifcsdef{if@openright}%
            {%
                \if@openright
                \cleardoublepage
            }%
        }%
    }%
}

```

```

        \else
        \clearpage
        \fi
    }%
    {%
    \cleardoublepage
    }%
    }%
    }%
    {}%
}
\newcommand*\glsclearpage{\gls@doclearpage}
\newcommand*\@gls@toc}[2]{%
    \ifglstoc
        \ifglsnumberline
            \addcontentsline{toc}{#2}{\protect\numberline{#1}}%
        \else
            \addcontentsline{toc}{#2}{#1}%
        \fi
    \fi
}
\newcommand*\glsnoxindywarning}[1]{%
    \GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
}
\newcommand*\glsnomakeindexwarning}[1]{%
    \GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%
}
\ifglsxindy
    \edef\@xdyattributes{\string"default\string"}%
\fi
\ifglsxindy
    \edef\@xdyattributelist{}%
\fi
\ifglsxindy
    \def\@xdylocref{}
\fi
\newcommand*\@gls@ifinlist}[4]{%
    \def\@do@ifinlist##1,#1,##2\end@ifinlist{%
        \def\@gls@listsuffix{##2}%
        \ifx\@gls@listsuffix\@empty
            #4%
        \else
            #3%
        \fi
    }%
    \@do@ifinlist,#2,#1,\end@ifinlist
}
\ifglsxindy
    \newcommand*\@xdycounters{\glscounter}
    \newcommand*\GlsAddXdyCounters[1]{%

```

```

\@for\@gls@ctr:=#1\do{%
  \edef\@do@addcounter{%
    \noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
    {%
      \noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
        \noexpand\@gls@ctr}%
      }%
    }%
  \do@addcounter
}
}
\@onlypremakeg\GlsAddXdyCounters
\else
\newcommand*\GlsAddXdyCounters[1]{%
  \glsnoxindywarning\GlsAddXdyAttribute
}
\fi
\newcommand*\@disabled@glsaddxdycounters{%
  \PackageError{glossaries}{\string\GlsAddXdyCounters\space
can't be used after \string\GlsAddXdyAttribute}{Move all
occurrences of \string\GlsAddXdyCounters\space before the first
instance of \string\GlsAddXdyAttribute}%
}
\ifglsxindy
\newcommand*\@glsaddxdyattribute[2]{%
  \edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string" ^^J
\string"#2#1\string"}%
  \expandafter\toks@\expandafter{\@xdylocref}%
  \edef\@xdylocref{\the\toks@ ^^J}%
  (markup-locref
:open \string"\glstildechar n%
  \expandafter\string\csname glsX#2X#1\endcsname
  \string" ^^J
:close \string"\string" ^^J
:attr \string"#2#1\string"))%
  \expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
    \setentrycounter{##1}{#2}\csname #1\endcsname{##2}%
  }%
}
\newcommand*\GlsAddXdyAttribute[1]{%
  \ifx\@xdyattributelist\@empty
  \edef\@xdyattributelist{#1}%
  \else
  \edef\@xdyattributelist{\@xdyattributelist,#1}%
  \fi
\@for\@this@counter:=\@xdycounters\do{%
  \protected@edef\gls@do@addxdyattribute{%
    \noexpand\@glsaddxdyattribute{#1}{\@this@counter}%
  }
  \gls@do@addxdyattribute
}

```

```

    }%
    \let\GlsAddXdyCounters\@disabled@gl saddx dycounters
  }
  \@onlypremakeg\GlsAddXdyAttribute
\else
  \newcommand*\GlsAddXdyAttribute[1]{%
    \gl snoxindywarning\GlsAddXdyAttribute}
\fi
\ifgl sxindy
\newcommand*{\@gl s@addpredefinedattributes}{%
  \GlsAddXdyAttribute{gl snumberformat}
  \GlsAddXdyAttribute{textrm}
  \GlsAddXdyAttribute{textsf}
  \GlsAddXdyAttribute{texttt}
  \GlsAddXdyAttribute{textbf}
  \GlsAddXdyAttribute{textmd}
  \GlsAddXdyAttribute{textit}
  \GlsAddXdyAttribute{textup}
  \GlsAddXdyAttribute{textsl}
  \GlsAddXdyAttribute{textsc}
  \GlsAddXdyAttribute{emph}
  \GlsAddXdyAttribute{gl shypernumber}
  \GlsAddXdyAttribute{hyperm}
  \GlsAddXdyAttribute{hypersf}
  \GlsAddXdyAttribute{hypertt}
  \GlsAddXdyAttribute{hyperbf}
  \GlsAddXdyAttribute{hypermd}
  \GlsAddXdyAttribute{hyperit}
  \GlsAddXdyAttribute{hyperup}
  \GlsAddXdyAttribute{hypersl}
  \GlsAddXdyAttribute{hypersc}
  \GlsAddXdyAttribute{hyperemph}
  \GlsAddXdyAttribute{gl signore}
}
\else
  \let\@gl s@addpredefinedattributes\relax
\fi
\def\@xdyuseralphabets{}
\ifgl sxindy
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \edef\@xdyuseralphabets{%
      \@xdyuseralphabets ^~J
      (define-alphabet "#1" (#2))}}
\else
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \gl snoxindywarning\GlsAddXdyAlphabet}
\fi
\ifgl sxindy
  \newcommand*{\@gl s@xdy@locationlist}{%
    roman-page-numbers,%

```

```

    Roman-page-numbers,%
    arabic-page-numbers,%
    alpha-page-numbers,%
    Alpha-page-numbers,%
    Appendix-page-numbers,%
    arabic-section-numbers%
}
\protected@edef\@gls@roman{\@roman{0}\string"
  \string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@gls@roman
\edef\@gls@tmp{\string" \string"roman-numbers-lowercase\string"
  :sep \string"}%
\@onelevel@sanitize\@gls@tmp
\ifx\@gls@tmp\@gls@roman
  \expandafter
    \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
      \string"roman-numbers-lowercase\string"%
    }%
\else
  \expandafter
    \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
      :sep \string"\@gls@roman\string"%
    }%
\fi
\expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
  \string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
  \string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
  \string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
  \string"ALPHA\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
  \string"ALPHA\string"
  :sep \string"\@gls@alphacompositor\string"
  \string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
  \string"arabic-numbers\string"
  :sep \string"\glscompositor\string"
  \string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglxindy

```

```

\newcommand\@xdycrossrefhook{}
\fi
\ifglxindy
\newcommand*\GlsAddXdyLocation}[3][{}]{%
\def\@gls@tmp{#1}%
\ifx\@gls@tmp\@empty
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep \string"{}\glsopenbrace\string" #3
:sep \string"\glsclosebrace\string"))
}%
\else
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep "\glsopenbrace"
#1
:sep "\glsclosebrace\glsopenbrace" #3
:sep "\glsclosebrace"))
}%
\fi
\edef\@xdyuserlocationnames{%
\@xdyuserlocationnames^^J\space\space\space
\string"#2\string"}%
}
\@onlypremakeg\GlsAddXdyLocation
\else
\newcommand*\GlsAddXdyLocation}[2][{}]{%
\glsnoxywarning\GlsAddXdyLocation}
\fi
\ifglxindy
\def\@xdylocationclassorder{^^J\space\space\space
\string"roman-page-numbers\string"^^J\space\space\space
\string"arabic-page-numbers\string"^^J\space\space\space
\string"arabic-section-numbers\string"^^J\space\space\space
\string"alpha-page-numbers\string"^^J\space\space\space
\string"Roman-page-numbers\string"^^J\space\space\space
\string"Alpha-page-numbers\string"^^J\space\space\space
\string"Appendix-page-numbers\string"
\@xdyuserlocationnames^^J\space\space\space
\string"see\string"}
\fi
\ifglxindy
\newcommand*\GlsSetXdyLocationClassOrder[1][{}]{%
\def\@xdylocationclassorder{#1}}
\else
\newcommand*\GlsSetXdyLocationClassOrder[1][{}]{%
\glsnoxywarning\GlsSetXdyLocationClassOrder}

```

```

\fi
\ifglxindy
  \def\@xdysortrules{}
\fi
\ifglxindy
  \newcommand*\GlsAddSortRule[2]{%
    \expandafter\toks@\expandafter{\@xdysortrules}%
    \protected@edef\@xdysortrules{\the\toks@ ^^J
      (sort-rule \string"#1\string" \string"#2\string")}%
    }
\else
  \newcommand*\GlsAddSortRule[2]{%
    \glsnoxywarning\GlsAddSortRule}
\fi
\ifglxindy
  \def\@xdyrequiredstyles{tex}
\fi
\ifglxindy
  \newcommand*\GlsAddXdyStyle[1]{%
    \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
  \newcommand*\GlsAddXdyStyle[1]{%
    \glsnoxywarning\GlsAddXdyStyle}
\fi
\ifglxindy
  \newcommand*\GlsSetXdyStyles[1]{%
    \edef\@xdyrequiredstyles{#1}}
\else
  \newcommand*\GlsSetXdyStyles[1]{%
    \glsnoxywarning\GlsSetXdyStyles}
\fi
\newcommand*\findrootlanguage{-}
\def\@xdylanguage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
    \ifglossaryexists{#1}{%
      \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
    }{%
      \PackageError{glossaries}{Can't set language type for
        glossary type '#1' --- no such glossary}{%
        You have specified a glossary type that doesn't exist}}
\else
  \newcommand*\GlsSetXdyLanguage[2][ ]{%
    \glsnoxywarning\GlsSetXdyLanguage}
\fi
\def\@gls@codepage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyCodePage[1]{%
    \renewcommand*\@gls@codepage{#1}%
  }

```



```

\AtBeginDocument{%
  \ifx\gls@codepage@empty
    \ifpackageloaded{fontspec}{\def\gls@codepage{utf8}}{}%
  \fi
}
\else
  \newcommand*\GlsSetXdyCodePage[1]{%
    \glsnoxindywarning\GlsSetXdyCodePage}
\fi
\ifglsxindy
  \ifgls@xindy@glsnumbers
    \def\@xdylettergroups{(define-letter-group
      \string\glsnumbers\string^^J\space\space\space
      :prefixes (\string"0\string" \string"1\string"
        \string"2\string" \string"3\string" \string"4\string"
        \string"5\string" \string"6\string" \string"7\string"
        \string"8\string" \string"9\string")^^J\space\space\space
      \@xdynumbergrouporder)}
    \else
      \def\@xdylettergroups{}
    \fi
\fi
\newcommand*\GlsAddLetterGroup[2]{%
  \expandafter\toks@\expandafter{\@xdylettergroups}%
  \protected@edef\@xdylettergroups{\the\toks@^^J%
    (define-letter-group \string"#1\string"^^J\space\space\space#2)}%
  }%
\newcommand*\forallglossaries[3][\@glo@types]{%
  \@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%
}

\newcommand*\forallacronyms[2]{%
  \@for#1:=\@glsacronymlists\do{\ifx#1\@empty\else#2\fi}%
}
\newcommand*\forglentries[3][\glsdefaulttype]{%
  \protected@edef\@glo@list{\csname glo@list@#1\endcsname}%
  \@for#2:=\@glo@list\do
  {%
    \ifdefempty{#2}{#3}%
  }%
}
\newcommand*\forallglentries[3][\@glo@types]{%
  \expandafter\forallglossaries\expandafter[#1]{\@@this@glo@}%
  {%
    \forglentries[\@@this@glo@]{#2}{#3}%
  }%
}
\newcommand\ifglossaryexists{%
  \@ifstar\s@ifglossaryexists\@ifglossaryexists
}

```

```

\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{@glotype@#1@out}{#3}{#2}%
}
\newcommand{\s@ifglossaryexists}[3]{%
  \ifcsundef{glolist@#1}{#3}{#2}%
}
\newcommand*{\glsdetoklabel}[1]{#1}
\newcommand{\ifglsentryexists}[3]{%
  \ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}
\newcommand*{\ifglsused}[3]{%
  \ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}
\newcommand{\glsdoifexists}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
}
\newcommand{\glsdoifnoexists}[2]{%
  \ifglsentryexists{#1}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
      been defined}{#2}%
}
}
\newcommand{\glsdoifexistsorwarn}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}%
  }%
}
\newcommand{\glsdoifexistsordo}[3]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
  #3%
}%
}
\newcommand{\doifglossarynoexistsordo}[3]{%
  \ifglossaryexists*{#1}%
  {%
    \PackageError{glossaries}{Glossary type ‘#1’ already exists}{#2}%
  }%
  #3%
}
\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@glstetoklabel{#1}@short}%
    \ifdefempty\@glo@short
      {#3}%
    {%
      \ifdequal\@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%
            }%
          {%
            \ifdequal\@glo@thisvalue\@gls@default@value
              {%
                #4%
              }%
            {%
              \let\glscurrentfieldvalue\@glo@thisvalue
                #3%
              }%
            }%
          }%
        }%
      {%
        \@gls@fetchfield{\@gls@thisfield}{#1}%
        \ifdef\@gls@thisfield
          {%
            \letcs{\@glo@thisvalue}{glo@glstetoklabel{#2}@#1}%
            \ifdefempty\@glo@thisvalue
              {%
                #4%
              }%
            {%
              \ifdequal\@glo@thisvalue\@gls@default@value
                {%
                  #4%
                }%
              {%
                \let\glscurrentfieldvalue\@glo@thisvalue
                  #3%
                }%
              }%
            }%
          }%
        }%
      }%
    }%
  }

```



```

\expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
\expandafter\@gls@forbidtextext\csname @glotype@#2@log\endcsname
\expandafter\@gls@forbidtextext\csname @glotype@#2@in\endcsname
\expandafter\@gls@forbidtextext\csname @glotype@#2@out\endcsname
\expandafter\def\csname @glotype@#2@title\endcsname{#5}%
\@gls@provide@newglossary
\protected@write\@auxout{}\string\@newglossary{#2}{#1}{#3}{#4}}%
\ifcsundef{gls@#2@entryfmt}%
{%
  \defglsentryfmt[#2]{\glsentryfmt}%
}%
}%
\@gls@defsortcount{#2}%
\@ifnextchar[{\@gls@setcounter{#2}}%
  {\@gls@setcounter{#2}[\glscounter]}%
}%
{%
  \gls@gobbleopt
}%
}
\newcommand*\altnewglossary}[3]{%
  \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
}
\@onlypreamble{\newglossary}
\@onlypremakeg\newglossary
\newcommand*\@newglossary}[4]{
\def\@gls@setcounter#1[#2]{%
  \expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
  \ifglxindy
    \GlsAddXdyCounters{#2}%
  \fi
}
\newcommand*\@gls@getcounter}[1]{%
  \csname @glotype@#1@counter\endcsname
}
\glsdefmain
\@gls@do@acronymsdef
\@gls@do@symbolsdef
\@gls@do@numbersdef
\@gls@do@indexdef
\newcommand*\@newignoredglossary}[1]{%
  \ifdefempty\@ignored@glossaries
  {%
    \protected@edef\@ignored@glossaries{#1}%
  }%
  {%
    \protected@eappto\@ignored@glossaries{,#1}%
  }%
  \csgdef{glolist@#1}{,}%
  \ifcsundef{gls@#1@entryfmt}%

```

```

{%
  \defglsentryfmt[#1]{\glsentryfmt}%
}%
{}%
\ifdefempty\@gls@nohyperlist
{%
  \renewcommand*\@gls@nohyperlist{#1}%
}%
{%
  \protected@eappto\@gls@nohyperlist{, #1}%
}%
}
\newcommand*\@ignored@glossaries{}
\newcommand*\@ifignoredglossary[3]{%
  \protected@edef\@gls@igtype{#1}%
  \expandafter\DTLifinlist\expandafter
  {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
}
\define@key{glossentry}{name}{%
\def\@glo@name{#1}%
}
\define@key{glossentry}{description}{%
\def\@glo@desc{#1}%
}
\define@key{glossentry}{descriptionplural}{%
\def\@glo@descplural{#1}%
}
\define@key{glossentry}{sort}{%
\def\@glo@sort{#1}}
\define@key{glossentry}{text}{%
\def\@glo@text{#1}%
}
\define@key{glossentry}{plural}{%
\def\@glo@plural{#1}%
}
\define@key{glossentry}{first}{%
\def\@glo@first{#1}%
}
\define@key{glossentry}{firstplural}{%
\def\@glo@firstplural{#1}%
}
\newcommand*\@gls@default@value{\relax}
\define@key{glossentry}{symbol}{%
\def\@glo@symbol{#1}%
}
\define@key{glossentry}{symbolplural}{%
\def\@glo@symbolplural{#1}%
}
\define@key{glossentry}{type}{%
\def\@glo@type{#1}}

```

```

\define@key{glossentry}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
}%
\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@noname}{%
  \PackageError{glossaries}{name key required in
  \string\newglossaryentry\space for entry '\@glo@label'}{You
  haven't specified the entry name}
}

```

```

\newcommand*\@glsnodesc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '\@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}%
\newcommand*{\@glsdefaultplural}{}
\newcommand*{\@gls@missingnumberlist}[1]{%
  ??%
  \ifglssavenumberlist
  \GlossariesWarning{Missing number list for entry '#1'.
    Maybe makeglossaries + rerun required}%
  \else
  \PackageError{glossaries}%
  {Package option 'savenumberlist=true' required}%
  {%
    You must use the 'savenumberlist' package option
    to reference location lists.%
  }%
  \fi
}
\newcommand*{\@glsdefaultsort}{\@glo@name}
\newcount\gls@level
\newcommand{\@gls@noexpand@field}[3]{%
  \expandafter\global\expandafter
  \let\csname glo@#1@#2\endcsname#3%
}
\newcommand{\@gls@noexpand@fields}[4]{%
  \ifcsdef{gls@assign@#3@field}
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
    }%
    {%
      \csuse{gls@assign@#3@field}{#2}{#4}%
    }%
  }%
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \@gls@noexpand@field{#2}{#3}{\@gls@value}%
    }%
    {%
  }
}

```

```

        \@@gls@noexpand@field{#2}{#3}{#4}%
    }%
}
\newcommand{\@@gls@expand@field}[3]{%
\expandafter
\protected@xdef\csname glo@#1@#2\endcsname{#3}%
}
\newcommand{\@gls@expand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
{%
\@@gls@expand@field{#2}{#3}{#4}%
}%
{%
\csuse{gls@assign@#3@field}{#2}{#4}%
}%
}%
}%
\ifdefequal{#4}{\@gls@default@value}%
{%
\@@gls@expand@field{#2}{#3}{#1}%
}%
{%
\@@gls@expand@field{#2}{#3}{#4}%
}%
}%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
\def\@gls@tmp{#1}%
\ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*\{gls@expandfields}{%
\let\gls@assign@field\@gls@expand@fields
}
\newcommand*\{gls@noexpandfields}{%
\let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
\glsdoifnoexists{#1}%

```

```

    {%
      \gls@defglossaryentry{#1}{#2}%
    }%
  }
\newcommand*\gls@defdocnewglossaryentry{%
  \let\newglossaryentry\new@glossaryentry
}
\newrobustcmd{\provideglossaryentry}[2]{%
  \ifglsentryexists{#1}%
  {%
    {%
      \gls@defglossaryentry{#1}{#2}%
    }%
  }
}
\@onlypreamble{\provideglossaryentry}
\newcommand*\gls@glsdefs@labellist{}
\newcommand*\gls@glsdefs@addlabel}[1]{\listxadd{\gls@glsdefs@labellist}{#1}}
\newcommand*\gls@docdef@labellist{}
\newcommand*\gls@docdef@addlabel}[1]{\listxadd{\gls@docdef@labellist}{#1}}
\newrobustcmd{\new@glossaryentry}[2]{%
  \ifundef\@gls@deffile
  {%
    \global\newwrite\@gls@deffile
    \immediate\openout\@gls@deffile=\jobname.glsdefs
  }%
  {%
    \ifglsentryexists{#1}%
    {%
      \xifinlist{#1}{\gls@docdef@labellist}%
      {%
        \GlossariesWarning{Entry ‘#1’ has already been defined in the
          document environment (ignoring new definition)}%
      }%
      {%
        \xifinlist{#1}{\gls@glsdefs@labellist}%
        {%
          \gls@defglossaryentry{#1}{#2}%
          \gls@docdef@addlabel{#1}%
          \@gls@writedef{#1}%
        }%
        {%
          \GlossariesWarning{Entry ‘#1’ has already been defined in
            the preamble (in which case remove the duplicate definition)
            or you have just updated to glossaries.sty
            v4.47 (in which case delete ‘\jobname.glsdefs’)}%
        }%
      }%
    }%
  }%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}

```

```

        \gls@docdef@addlabel{#1}%
        \@gls@writedef{#1}%
    }%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*\gls@begindocdefs{%
    \gls@enablesavenonumberlist
    \edef\@gls@restreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
    \makeatletter
    \InputIfFileExists{\jobname.glsdefs}{-}{-}%
    \@gls@restreat
    \undef\@gls@restreat
    \gls@defdocnewglossaryentry
}
\newcommand*\glsdefs@newdocentry}[2]{%
    \ifglsentryexists{#1}%
    {\GlossariesWarning{entry ‘#1’ has already been defined in the preamble
        (if you have moved the definition there from the document
        environment, delete \jobname.glsdefs)}}%
    {%
        \gls@glsdefs@addlabel{#1}%
        \gls@defglossaryentry{#1}{#2}%
    }%
}
\newcommand*\@gls@writedef}[1]{%
    \immediate\write\@gls@deffile
    {%
        \string\glsdefs@newdocentry{\glsdetoklabel{#1}}\glspercentchar^^J%
        \expandafter\@gobble\string\{\glspercentchar%
    }%
    \@for\@gls@map:=\@gls@keymap\do
    {%
        \letcs\glo@value{glo@\glsdetoklabel{#1}}\expandafter\@secondoftwo\@gls@map}%
        \ifdef\glo@value
        {%
            \@onelevel@sanitize\glo@value
            \immediate\write\@gls@deffile
            {%
                \expandafter\@firstoftwo\@gls@map
                =\expandafter\@gobble\string\{\glo@value\expandafter\@gobble\string\},%
                \glspercentchar
            }%
        }%
    }%
}
\gls@writedefhook
\immediate\write\@gls@deffile
{%
    \glspercentchar^^J%
}

```

```

    \expandafter\@gobble\string\}\glspercentchar^^J%
  }%
}
\newcommand*{\@gls@keymap}{%
  {name}{name},%
  {sort}{sortvalue},% unescaped sort value
  {type}{type},%
  {first}{first},%
  {firstplural}{firstpl},%
  {text}{text},%
  {plural}{plural},%
  {description}{desc},%
  {descriptionplural}{descplural},%
  {symbol}{symbol},%
  {symbolplural}{symbolplural},%
  {user1}{useri},%
  {user2}{userii},%
  {user3}{useriii},%
  {user4}{useriv},%
  {user5}{userv},%
  {user6}{uservi},%
  {long}{long},%
  {longplural}{longpl},%
  {short}{short},%
  {shortplural}{shortpl},%
  {counter}{counter},%
  {parent}{parent}%
}
\newcommand*{\@gls@fetchfield}[2]{%
  \edef\@gls@thisval{#2}%
  \@for\@gls@map:=\@gls@keymap\do{%
    \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
    \ifdefequal{\@this@key}{\@gls@thisval}%
    {%
      \edef#1{\expandafter\@secondoftwo\@gls@map}%
      \@endfortrue
    }%
  }%
}%
}
\newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
\newcommand*{\@sglsaddstoragekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
}%
}

```

```

\@glsaddstoragekey{#1}%
}
\newcommand*{\@glsaddstoragekey}[3]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{\@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
}%
{%
\PackageError{glossaries}{Key ‘#1’ already exists}{}%
}%
}
\newcommand*{\glsaddkey}{\@ifstar\@sglsaddkey\@glsaddkey}
\newcommand*{\@sglsaddkey}[1]{%
\key@ifundefined{glossentry}{#1}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname gls@assign@#1@field\endcsname}[2]{%
\@gls@expand@field{##1}{#1}{##2}%
}%
}%
{}%
\@glsaddkey{#1}%
}
\newcommand*{\@glsaddkey}[7]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{\@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
\newcommand*{#4}[1]{\@Gls@entry@field{##1}{#1}}%
\ifcsdef{@gls@user@#1@}%
{%
\PackageError{glossaries}%
{Can’t define ‘\string#5’ as helper command
‘\expandafter\string\csname @gls@user@#1@\endcsname’ already exists}%
{}%
}%
{}%
}

```

```

\expandafter\newcommand\expandafter*\expandafter
  {\csname @gls@user@#1\endcsname}[2][]{%
    \new@ifnextchar[%
      {\csuse{@gls@user@#1@}{##1}{##2}}%
      {\csuse{@gls@user@#1@}{##1}{##2}[]}}%
\csdef{@gls@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{#3{##2}##3}%
}%
\newrobustcmd*{#5}{%
  \expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
}%
\ifcsdef{@Gls@user@#1@}{%
  {%
    \PackageError{glossaries}{%
      {Can't define '\string#6' as helper command
        '\expandafter\string\csname @Gls@user@#1\endcsname' already exists}}%
    }%
  }%
  {%
    \expandafter\newcommand\expandafter*\expandafter
      {\csname @Gls@user@#1\endcsname}[2][]{%
        \new@ifnextchar[%
          {\csuse{@Gls@user@#1@}{##1}{##2}}%
          {\csuse{@Gls@user@#1@}{##1}{##2}[]}}%
\csdef{@Gls@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{#4{##2}##3}%
}%
\newrobustcmd*{#6}{%
  \expandafter\@gls@hyp@opt\csname @Gls@user@#1\endcsname}%
}%
\ifcsdef{@GLS@user@#1@}{%
  {%
    \PackageError{glossaries}{%
      {Can't define '\string#7' as helper command
        '\expandafter\string\csname @GLS@user@#1\endcsname' already exists}}%
    }%
  }%
  {%
    \expandafter\newcommand\expandafter*\expandafter
      {\csname @GLS@user@#1\endcsname}[2][]{%
        \new@ifnextchar[%
          {\csuse{@GLS@user@#1@}{##1}{##2}}%
          {\csuse{@GLS@user@#1@}{##1}{##2}[]}}%
\csdef{@GLS@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{\mfirstucMakeUppercase{#3{##2}##3}}%
}%
\newrobustcmd*{#7}{%
  \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
}%
}%

```



```

    {%
      \PackageError{glossaries}{Key ‘#1’ already exists}{-}%
    }%
  }
\newcommand{\glsfieldxdef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \protected@csxdef{glo@\@glo@label @#2}{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{-}%
    }%
  }%
}
\newcommand{\glsfielddedef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \protected@csedef{glo@\@glo@label @#2}{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{-}%
    }%
  }%
}
\newcommand{\glsfieldgdef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{-}%
    }%
  }%
}
\newcommand{\glsfieldddef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
    }%
  }%
}

```

```

}%
{%
  \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
}%
}%
}
\newcommand{\glsfieldfetch}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \letcs#3{glo@\@glo@label @#2}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand*{\glsunexpandedfieldvalue}[2]{%
  \ifcsname glo@\glsdetoklabel{#1}@#2\endcsname
  \expandafter\expandonce\csname glo@\glsdetoklabel{#1}@#2\expandafter\endcsname
  \fi
}
\newcommand{\ifglsfieldeq}[5]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand{\ifglsfielddfeq}[5]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \expandafter\ifdefstrequal
    \csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}%
}

```

```

}
\newcommand{\ifglsfieldcseq}[5]{%
\glsdoifexists{#1}%
{%
\protected@edef\glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\glo@label @#2}%
{%
\ifcsstrequal{glo@\glo@label @#2}{#3}{#4}{#5}%
}%
{%
\PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
}%
}%
}
\newcommand*{\glswritedefhook}{}
\newcommand*{\gls@assign@desc}[1]{%
\gls@assign@field{#1}{desc}{\glo@desc}%
\gls@assign@field{\glo@desc}{#1}{descplural}{\glo@descplural}%
}
\newcommand{\longnewglossaryentry}[3]{%
\glsdoifnoexists{#1}%
{%
\bgroup
\let\org@newglossaryentryprehook\@newglossaryentryprehook
\long\def\@newglossaryentryprehook{%
\long\def\glo@desc{#3\leavevmode\nskip\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@default@value
\let\glo@type\gls@default@value

```

```

\let@glo@symbol@gls@default@value
\let@glo@symbolplural@gls@default@value
\let@glo@text@gls@default@value
\let@glo@plural@gls@default@value
\let@glo@first@gls@default@value
\let@glo@firstplural@gls@default@value
\let@glo@sort@gls@default@value
\let@glo@counter@gls@default@value
\def@glo@see{}%
\def@glo@parent{}%
\def@glo@prefix{}%
@gls@initnonumberlist
\def@glo@useri{}%
\def@glo@userii{}%
\def@glo@useriii{}%
\def@glo@useriv{}%
\def@glo@userv{}%
\def@glo@uservi{}%
\def@glo@short{}%
\def@glo@shortpl{}%
\def@glo@long{}%
\def@glo@longpl{}%
@newglossaryentryprehook
\setkeys{glossentry}{#2}%
\ifundef@glsdefaulttype
{%
  \PackageError{glossaries}%
  {No default glossary type (have you used ‘nomain’ by mistake?)}%
  {If you use package option ‘nomain’ you must define
   a new glossary before you can define entries}%
}%
{}%
\gls@assign@field{\glsdefaulttype}{\glo@label}{type}{\glo@type}%
\protected@edef@glo@type{\glsentrytype{\glo@label}}%
\ifcsundef{glo@list@\glo@type}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘\glo@type’ has not been defined}%
  {You need to define a new glossary type, before making entries
   in it}%
}%
{}%
\ifignoredglossary@glo@type
{%
  \ifx@glo@desc@glsnodesc
    \let@glo@desc@empty
  \fi
}%
{}%
{}%

```

```

\protected@edef\glo@list@{\csname glo@list@\@glo@type\endcsname}%
\expandafter\protected@xdef\csname glo@list@\@glo@type\endcsname{%
  \glo@list@{\@glo@label},}%
}%
\gls@level=0\relax
\ifx\@glo@parent\@empty
  \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
\else
  \ifdefequal\@glo@label\@glo@parent%
  {%
    \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
    \def\@glo@parent{}%
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
  }%
  {%
    \ifglsentryexists{\@glo@parent}%
    {%
      \expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
        \@glo@parent}%
      \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
      \advance\gls@level by 1\relax
      \ifx\@glo@name\@gls@noname
        \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\gls@pluralsuffix}%

```

```

        {\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @text\endcsname}%
    {\@glo@label}{first}{\@glo@first}%
\ifx\@glo@first\@gls@default@value
    \expandafter\gls@assign@field\expandafter
        {\csname glo@\@glo@label @plural\endcsname}%
        {\@glo@label}{firstpl}{\@glo@firstplural}%
\else
    \expandafter\gls@assign@field\expandafter
        {\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
        {\@glo@label}{firstpl}{\@glo@firstplural}%
\fi
\ifcsundef{@glotype@\@glo@type @counter}%
{%
    \def\@glo@defaultcounter{\glscounter}%
}%
{%
\letcs\@glo@defaultcounter{@glotype@\@glo@type @counter}%
}%
\gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
\gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
\gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
\gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
\gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
\gls@assign@field{}{\@glo@label}{userv}{\@glo@userv}%
\gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
\gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
\gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
\gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
\gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%
\ifx\@glo@name\@glsnoname
    \@glsnoname
    \let\@glo@name\@gls@default@value
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%
\ifcsundef{glo@\@glo@label @numberlist}%
{%
    \csxdef{glo@\@glo@label @numberlist}{%
        \noexpand\@gls@missingnumberlist{\@glo@label}}%
}%
{%
}%
\@gls@storenonumberlist{\@glo@label}%
\def\@glo@@desc{\@glo@first}%
\ifx\@glo@desc\@glo@@desc
    \let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@glsnodesc
    \@glsnodesc
    \let\@glo@desc\@gls@default@value

```

```

\fi
\gls@assign@desc{\@glo@label}%
\gls@defsort{\@glo@type}{\@glo@label}%
\def\@glo@@symbol{\@glo@text}%
\ifx\@glo@symbol\@glo@@symbol
  \let\@glo@symbol\@glo@text
\fi
\gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
\expandafter
  \gls@assign@field\expandafter
  {\csname glo@\@glo@label @symbol\endcsname}
  {\@glo@label}{symbolplural}{\@glo@symbolplural}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagfalse\endcsname{%
  \noexpand\global
  \noexpand\let\expandafter\noexpand
  \csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagtrue\endcsname{%
  \noexpand\global
  \noexpand\let\expandafter\noexpand
  \csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
}%
\csname glo@\@glo@label @flagfalse\endcsname
\@glo@autosee
\ifignoredglossary\@glo@type
{%
  \csdef{glo@\@glo@label @index}{}%
}
{%
  \do@glo@storeentry{\@glo@label}%
}%
\@newglossaryentry@defcounters
\@newglossaryentryposthook
}
\newcommand*{\@glo@autosee}{%
  \ifdefvoid\@glo@see{%
    {%
      \protected@edef\@do@glssee{%
        \noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
        \noexpand\expandafter\noexpand\@glssee\noexpand\@glo@list{\@glo@label}}%
      \do@glssee
    }%
    \@glo@autoseehook
  }%
}
\newcommand*{\@glo@autoseehook}{}
\newcommand*{\@newglossaryentryprehook}{}
\newcommand*{\@newglossaryentryposthook}{}
\newcommand*{\@newglossaryentry@defcounters}{}
\newcommand*{\glsmoveentry}[2]{%
  \protected@edef\@glo@thislabel{\glsdetoklabel{#1}}%

```



```

\else
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
\csname glo@%glo@parent @index\endcsname\@gls@levelchar
\@glo@sort\@gls@actualchar\@glo@prefix
\@glossarysubentryfield
{\csname glo@#1@level\endcsname}\@glo@esclabel}%
}%
\fi
\fi
}
\AtBeginDocument{%
\@ifpackageloaded{amsmath}%
{\let\gls@ifnotmeasuring\@gls@ifnotmeasuring}%
{}}%
}
\newcommand*{\@gls@ifnotmeasuring}[1]{%
\ifmeasuring@
\else
#1%
\fi
}
\newcommand*\gls@ifnotmeasuring[1]{#1}
\def\@gls@patchtabularx#1\hbox#2#3!!{%
\def\TX@trial##1{#1\hbox{\let\glsunset\@gobble#2}#3}%
}
\newcommand*\glspatchtabularx{%
\ifdef\TX@trial
{%
\expandafter\@gls@patchtabularx\TX@trial{##1}!!%
\let\glspatchtabularx\relax
}%
{}}%
}
\newcommand*{\glsreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glsreset{#1}%
}%
}%
}
\newcommand*{\glslocalreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glslocalreset{#1}%
}%
}%
}

```

```

}
\newcommand*\glsunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glsunset{#1}%
    }%
  }%
}
\newcommand*\glslocalunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glslocalunset{#1}%
    }%
  }%
}
\newcommand*\@glslocalunset{\@glslocalunset}
\newcommand*\@glslocalunset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
}
\newcommand*\@glsunset{\@glsunset}
\newcommand*\@glsunset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
}
\newcommand*\@glslocalreset{\@glslocalreset}
\newcommand*\@glslocalreset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
}
\newcommand*\@glsreset{\@glsreset}
\newcommand*\@glsreset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
}
\newcommand*\glsresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsreset{\@glsentry}%
  }%
}
\newcommand*\glslocalresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalreset{\@glsentry}%
  }%
}
\newcommand*\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%

```

```

    \glsunset{\@glsentry}%
  }%
}
\newcommand*\glslocalunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalunset{\@glsentry}%
  }%
}
\newcommand*\@newglossaryentry@defcounters{%
  \csdef{glo@\@glo@label @currcount}{0}%
  \csdef{glo@\@glo@label @prevcount}{0}%
}
\newcommand*\glsenableentrycount{%
  \let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
  \renewcommand*\gls@defdocnewglossaryentry{%
    \renewcommand*\newglossaryentry[2]{%
      \PackageError{glossaries}{\string\newglossaryentry\space
        may only be used in the preamble when entry counting has
        been activated}{If you use \string\glsenableentrycount\space
        you must place all entry definitions in the preamble not in
        the document environment}%
    }%
  }%
}
\newcommand*\glsentrycurrcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
  {0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*\glsentryprevcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
  {0}{\@gls@entry@field{##1}{prevcount}}%
}%
\renewcommand*\@glsunset}[1]{%
  \@glsunset{##1}%
  \@gls@increment@currcount{##1}%
}%
\renewcommand*\@glslocalunset}[1]{%
  \@glslocalunset{##1}%
  \@gls@local@increment@currcount{##1}%
}%
\renewcommand*\@glsreset}[1]{%
  \@glsreset{##1}%
  \csgdef{glo@\glsdetoklabel{##1}@currcount}{0}%
}%
\renewcommand*\@glslocalreset}[1]{%
  \@glslocalreset{##1}%
  \csdef{glo@\glsdetoklabel{##1}@currcount}{0}%
}%
\def\@cgls###1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax

```

```

        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \@gls@{##1}{##2}[##3]%
    \fi
}%
\def\@cglsp@##1##2[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \@cglsp@{##1}{##2}[##3]%
    \fi
}%
\def\@Gls@##1##2[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \@Gls@{##1}{##2}[##3]%
    \fi
}%
\def\@Glspl@##1##2[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \@Glspl@{##1}{##2}[##3]%
    \fi
}%
}
\AtEndDocument{\@gls@write@entrycounts}%
\renewcommand*{\@gls@entry@count}[2]{%
    \csdef{glo@\glsdetoklabel{##1}@prevcount}{##2}%
}%
\let\glsetenableentrycount\relax
}
\@onlypreamble\glsetenableentrycount
\newcommand*{\@gls@increment@currcount}[1]{%
    \csxdef{glo@\glsdetoklabel{##1}@currcount}{%
        \number\numexpr\glsetcurrcount{##1}+1}%
}
\newcommand*{\@gls@local@increment@currcount}[1]{%
    \csedef{glo@\glsdetoklabel{##1}@currcount}{%
        \number\numexpr\glsetcurrcount{##1}+1}%
}
\newcommand*{\@gls@write@entrycounts}{%
    \immediate\write\@auxout
        {\string\providecommand*{\string\@gls@entry@count}[2]{}}%
    \forallglsentries{\@glsentry}{%
        \ifglsused{\@glsentry}%

```

```

        {\immediate\write\@auxout
         {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}%
        }%
    }%
}
\newcommand*{\@gls@entry@count}[2]{%
\newrobustcmd*{\cglS}{\@gls@hyp@opt\@cglS}
\newcommand*{\@cglS}[2][]{%
  \new@ifnextchar[{\@cglS@{#1}{#2}}{\@cglS@{#1}{#2}[]}%
}
\def\@cglS@#1#2[#3]{%
  \GlossariesWarning{\string\cglS\space is defaulting to
    \string\gls\space since you haven't enabled entry counting}%
  \@cglS@{#1}{#2}[#3]%
}
\newcommand*{\cglSformat}[2]{%
  \ifglshaslong{#1}{\glsentrylong{#1}}{\glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cGlS}{\@gls@hyp@opt\@cGlS}
\newcommand*{\@cGlS}[2][]{%
  \new@ifnextchar[{\@cGlS@{#1}{#2}}{\@cGlS@{#1}{#2}[]}%
}
\def\@cGlS@#1#2[#3]{%
  \GlossariesWarning{\string\cGlS\space is defaulting to
    \string\Gls\space since you haven't enabled entry counting}%
  \@cGlS@{#1}{#2}[#3]%
}
\newcommand*{\cGlSformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cglSpl}{\@gls@hyp@opt\@cglSpl}
\newcommand*{\@cglSpl}[2][]{%
  \new@ifnextchar[{\@cglSpl@{#1}{#2}}{\@cglSpl@{#1}{#2}[]}%
}
\def\@cglSpl@#1#2[#3]{%
  \GlossariesWarning{\string\cglSpl\space is defaulting to
    \string\glspl\space since you haven't enabled entry counting}%
  \@cglSpl@{#1}{#2}[#3]%
}
\newcommand*{\cglSplformat}[2]{%
  \ifglshaslong{#1}{\glsentrylongpl{#1}}{\glsentryfirstplural{#1}}#2%
}
\newrobustcmd*{\cGlSpl}{\@gls@hyp@opt\@cGlSpl}
\newcommand*{\@cGlSpl}[2][]{%
  \new@ifnextchar[{\@cGlSpl@{#1}{#2}}{\@cGlSpl@{#1}{#2}[]}%
}
\def\@cGlSpl@#1#2[#3]{%
  \GlossariesWarning{\string\cGlSpl\space is defaulting to
    \string\GlSpl\space since you haven't enabled entry counting}%
  \@cGlSpl@{#1}{#2}[#3]%
}

```

```

}
\newcommand*{\cGlsplformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
}
\newcommand*{\loadglsentries}[2][\@gls@default]{%
  \let\@gls@default\glsdefaulttype
  \def\glsdefaulttype{#1}\input{#2}%
  \let\glsdefaulttype\@gls@default
}
\@onlypreamble{\loadglsentries}
\newcommand*{\glstextformat}[1]{#1}
\newcommand*{\glsentryfmt}{%
  \@gls@default@entryfmt\glsdisplayfirst\glsdisplay
}
\newcommand*{\@gls@default@entryfmt}[2]{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          #2{\glsentryplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
        }%
        {%
          #1{\glsentryfirstplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
          }%
        }%
      }%
    }%
  }%
  \ifglsused\glslabel
  {%
    \ifbool{glscompatible-3.07}%
    {%
      \protected@edef\@glo@etext{%
        #2{\glsentryplural{\glslabel}}%
          {\glsentrydescplural{\glslabel}}%
          {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
      \xmakefirstuc\@glo@etext
    }%
    {%
      #2{\Glsentryplural{\glslabel}}%
        {\glsentrydescplural{\glslabel}}%
        {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
    }%
  }%
}

```

```

{%
  \ifbool{glscompatible-3.07}%
  {%
    \protected@edef\@glo@etext{%
      #1{\glsentryfirstplural{\glslabel}}%
      {\glsentrydescplural{\glslabel}}%
      {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
    \xmakefirstuc\@glo@etext
  }%
  {%
    #1{\Glsentryfirstplural{\glslabel}}%
    {\glsentrydescplural{\glslabel}}%
    {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
  }%
}%
}%
{%
  \ifglsused\glslabel
  {%
    \mfirstucMakeUppercase{#2{\glsentryplural{\glslabel}}%
      {\glsentrydescplural{\glslabel}}%
      {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
  }%
  {%
    \mfirstucMakeUppercase{#1{\glsentryfirstplural{\glslabel}}%
      {\glsentrydescplural{\glslabel}}%
      {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
  }%
}%
}%
{%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      #2{\glsentrytext{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
    {%
      #1{\glsentryfirst{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
  }%
}%
{%
  \ifglsused\glslabel
  {%
    \ifbool{glscompatible-3.07}%
    {%

```



```

    {%
      #1{\glscustomtext}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{}}%
    }%
  }%
}
\newcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          \glsentryplural{\glslabel}\glsinsert
        }%
        {%
          \glsentryfirstplural{\glslabel}\glsinsert
        }%
      }%
    }%
    \ifglsused\glslabel
    {%
      \Glsentryplural{\glslabel}\glsinsert
    }%
    {%
      \Glsentryfirstplural{\glslabel}\glsinsert
    }%
  }%
  {%
    \ifglsused\glslabel
    {%
      \mfirstucMakeUppercase
      {\glsentryplural{\glslabel}\glsinsert}%
    }%
    {%
      \mfirstucMakeUppercase
      {\glsentryfirstplural{\glslabel}\glsinsert}%
    }%
  }%
}
  }%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      \glsentrytext{\glslabel}\glsinsert
    }%
  }%
}

```

```

        {%
        \glsentryfirst{\glslabel}\glsinsert
        }%
    }%
    {%
    \ifglsused\glslabel
    {%
    \Glsentrytext{\glslabel}\glsinsert
    }%
    {%
    \Glsentryfirst{\glslabel}\glsinsert
    }%
    }%
    {%
    \ifglsused\glslabel
    {%
    \mfirstucMakeUppercase{\glsentrytext{\glslabel}\glsinsert}%
    }%
    {%
    \mfirstucMakeUppercase{\glsentryfirst{\glslabel}\glsinsert}%
    }%
    }%
    }%
    }%
    {%
    \glscustomtext\glsinsert
    }%
}
\newcommand*{\glsgenacfmt}{%
\ifdefempty\glscustomtext
{%
\ifglsused\glslabel
{%
\glsifplural
{%
\glscapscase
{%
\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert
}%
{%
\acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
}%
{%
\mfirstucMakeUppercase
{\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert}%
}%
}%
}%
\glscapscase
{%

```



```

\protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
\xmakefirstuc\gls@text
}
\newcommand*\genplacrfullformat}[2]{%
  \glsentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glsentryshortpl{#1}})%
}
\newcommand*\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \xmakefirstuc\gls@text
}
\newcommand*\glsdisplayfirst}[4]{#1#4}
\newcommand*\glsdisplay}[4]{#1#4}
\newcommand*\defglsdisplay}[2][\glsdefaulttype]{%
  \GlossariesWarning{\string\defglsdisplay\space is now obsolete.^^J
  Use \string\defglsentryfmt\space instead}%
  \expandafter\def\csname gls@#1@display\endcsname##1##2##3##4{#2}%
  \edef\@gls@doentrydef{%
    \noexpand\defglsentryfmt[#1]{%
      \noexpand\ifcsdef{gls@#1@displayfirst}}%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\glsdisplayfirst}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
    }%
  }%
  \@gls@doentrydef
}
\newcommand*\defglsdisplayfirst}[2][\glsdefaulttype]{%
  \GlossariesWarning{\string\defglsdisplayfirst\space is now obsolete.^^J
  Use \string\defglsentryfmt\space instead}%
  \expandafter\def\csname gls@#1@displayfirst\endcsname##1##2##3##4{#2}%
  \edef\@gls@doentrydef{%
    \noexpand\defglsentryfmt[#1]{%
      \noexpand\ifcsdef{gls@#1@display}}%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\glsdisplay}%
      }%
    }%
  }%
  \@gls@doentrydef
}

```

```

    }%
  }%
}%
\@gls@doentrydef
}
\define@key{glslink}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
}%
{%
  \def\@gls@counter{#1}%
}%
}
\define@key{glslink}{format}{%
  \def\@glsnumberformat{#1}}
\define@boolkey{glslink}{hyper}[true]{}
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\define@boolkey{glslink}{local}[true]{}
\newcommand*{\glslinkvar}[3]{#1}
\newcommand*{\glsifhyper}[2]{%
  \glslinkvar{#1}{#2}{#1}%
  \GlossariesWarning{\string\glsifhyper\space is deprecated. Did
  you mean \string\glsifhyperon\space or \string\glslinkvar?}%
}
\newcommand*{\@gls@hyp@opt}[1]{%
  \let\glslinkvar\@firstofthree
  \let\@gls@hyp@opt@cs#1\relax
  \@ifstar{\s@gls@hyp@opt}%
  {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*{\s@gls@hyp@opt}[1] []{%
  \let\glslinkvar\@secondofthree
  \@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1] []{%
  \let\glslinkvar\@thirdofthree
  \@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
  \@gls@hyp@opt\@gls@@link
}
\newcommand*{\@gls@@link}[3] []{%
  \glsdoifexistsordo{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \@gls@link[#1]{#2}{#3}%
  }
}

```

```

    }{%
      \glstextformat{#3}%
    }%
  \glspostlinkhook
}
\newcommand*{\glspostlinkhook}{}
\newcommand*{@gls@link@checkfirsthyper}{%
  \ifglsused{\glslabel}%
  {%
  }%
  {%
    \gls@checkisacronymlist\glstype
    \ifglshyperfirst
      \ifglsisacronymlist
        \ifglsacrfootnote
          \KV@glslink@hyperfalse
        \fi
      \fi
    \else
      \KV@glslink@hyperfalse
    \fi
  }%
  \glslinkcheckfirsthyperhook
}
\newcommand*{\glslinkcheckfirsthyperhook}{}
\newcommand*{\glslinkpostsetkeys}{}
\newcommand{\glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}
\newcommand*{\do@glsdisablehyperinlist}{%
  \expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
  {\KV@glslink@hyperfalse}{}}%
}
\newcommand*{@gls@setdefault@glslink@opts}{}
\def\@gls@link[#1]#2#3{%
  \leavevmode
  \protected@edef\glslabel{\glsdetoklabel{#2}}%
  \def\@gls@link@opts{#1}%
  \let\@gls@link@label\glslabel
  \def\@glsnumberformat{\glsnumberformat}%
  \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  \protected@edef\glstype{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
  \@gls@setdefault@glslink@opts
  \do@glsdisablehyperinlist
  \do@gls@link@checkfirsthyper
  \setkeys{glslink}{#1}%
  \glslinkpostsetkeys
  \@gls@saveentrycounter
  \@gls@setsort{\glslabel}%
  \@do@wrglossary{#2}%
  \ifKV@glslink@hyper

```

```

        \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
    \else
        \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
    \fi
    \let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
}
\newcommand*{\glolinkprefix}{glo:}
\def\glsentrycounter{\glscounter}%
\newcommand*{\@gls@saveentrycounter}{%
    \def\@gls@Hcounter{}%
    \ifthenelse{\equal{\@gls@counter}{equation}}%
    {
        \ifcsundef{xatlevel@}%
        {%
            \edef\theglsentrycounter{\expandafter\noexpand
                \csname the\@gls@counter\endcsname}%
        }%
        {%
            \ifx\xatlevel@\@empty
                \edef\theglsentrycounter{\expandafter\noexpand
                    \csname the\@gls@counter\endcsname}%
            \else
                \savecounters@
                \advance\c@equation by 1\relax
                \edef\theglsentrycounter{\csname the\@gls@counter\endcsname}%
                \ifcsundef{theH\@gls@counter}%
                {%
                    \def\@gls@Hcounter{\theglsentrycounter}%
                }%
                {%
                    \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
                }%
                \protected@edef\theHglsentrycounter{\@gls@Hcounter}%
                \restorecounters@
            \fi
        }%
    }%
    {%
        \edef\theglsentrycounter{\expandafter\noexpand
            \csname the\@gls@counter\endcsname}%
    }%
    \ifx\@gls@Hcounter\@empty
        \ifcsundef{theH\@gls@counter}%
        {%
            \def\theHglsentrycounter{\theglsentrycounter}%
        }%
        {%
            \protected@edef\theHglsentrycounter{\expandafter\noexpand
                \csname theH\@gls@counter\endcsname}%
        }%
    }%
}

```

```

\fi
}
\def\@set@glo@numformat#1#2#3#4{%
\expandafter\@glo@check@mkidxrangechar#3\@nil
\protected@edef#1{%
\@glo@prefix setentrycounter[#4]{#2}%
\expandafter\string\csname\@glo@suffix\endcsname
}%
\@gls@checkmkidxchars#1%
}
\def\@glo@check@mkidxrangechar#1#2\@nil{%
\if#1\relax
\def\@glo@prefix{()}%
\if\relax#2\relax
\def\@glo@suffix{glsnumberformat}%
\else
\def\@glo@suffix{#2}%
\fi
\else
\if#1\relax
\def\@glo@prefix{})%
\if\relax#2\relax
\def\@glo@suffix{glsnumberformat}%
\else
\def\@glo@suffix{#2}%
\fi
\else
\def\@glo@prefix{}\def\@glo@suffix{#1#2}%
\fi
\fi}
\newcommand*{\@gls@escbsdq}[1]{%
\def\@gls@checkedmkidx{}%
\let\gls@xdystring=#1\relax
\@onelevel@sanitize\gls@xdystring
\edef\do@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null}%
\do@gls@xdycheckbackslash
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\@for\@gls@tmp:=\gls@protected@pagefmts\do
{%
\edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\expandonce\@gls@tmp}%
\@onelevel@sanitize\@gls@sanitized@tmp
\edef\gls@dostubst{%
\noexpand\DTLsubstituteall\noexpand\gls@xdystring
{\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
}%
}

```



```

    \gls@dostsubst
  }%
  \let#1=\gls@xdyststring
}
\newcommand{\@gls@checkmkidxchars}[1]{%
  \ifglsxindy
    \@gls@escbsdq{#1}%
  \else
    \def\@gls@checkedmkidx{%
      \expandafter\@gls@checkquote#1\@nil""\null
      \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
      \def\@gls@checkedmkidx{%
        \expandafter\@gls@checkescquote#1\@nil""\null
        \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
        \def\@gls@checkedmkidx{%
          \expandafter\@gls@checkescactual#1\@nil\?\?\null
          \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
          \def\@gls@checkedmkidx{%
            \expandafter\@gls@checkactual#1\@nil??\null
            \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
            \def\@gls@checkedmkidx{%
              \expandafter\@gls@checkbar#1\@nil||\null
              \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
              \def\@gls@checkedmkidx{%
                \expandafter\@gls@checkescbar#1\@nil|||\null
                \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                \def\@gls@checkedmkidx{%
                  \expandafter\@gls@checklevel#1\@nil!!\null
                  \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                  \def\@gls@checkedmkidx{%
                    \expandafter\@gls@checkesclevel#1\@nil!\!\null
                    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                    \def\@gls@checkedmkidx{%
                      \edef\@do@gls@checkesc{%
                        \noexpand\@gls@checkesc\expandonce{#1}\noexpand\@nil
                        \@backslashchar\@backslashchar\noexpand\@nil
                      }%
                      \@do@gls@checkesc
                      \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                    \fi
                  \def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
                  \newtoks\@gls@tmpb
                  \def\@gls@checkquote#1"#2"#3\@nil{%
                    \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
                    \toks@=#1}%
                    \ifx\@nil#2\@nil
                    \ifx\@nil#3\@nil
                    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
                    \def\@gls@checkquote{\relax}%
                  \def\@gls@checkquote{\relax}%
                \fi
              \fi
            \fi
          \fi
        \fi
      \fi
    \fi
  }%
}

```

```

\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
\def\@@gls@checkquote{\@gls@checkquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@quotechar}%
\ifx\null#3\null
\def\@@gls@checkquote{\@gls@checkquote#2""\null}%
\else
\def\@@gls@checkquote{\@gls@checkquote#2"#3\null}%
\fi
\fi
\@@gls@checkquote
}
\def\@gls@checkescquote#1"#2"#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkescquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar
\@gls@quotechar\string"\@gls@quotechar}%
\def\@@gls@checkescquote{\@gls@checkescquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar}%
\ifx\null#3\null
\def\@@gls@checkescquote{\@gls@checkescquote#2""\null}%
\else
\def\@@gls@checkescquote{\@gls@checkescquote#2"#3\null}%
\fi
\fi
\@@gls@checkescquote
}
\def\@gls@checkescactual#1\?#2\?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkescactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar

```

```

        \@gls@quotechar\string"\@gls@actualchar}%
        \def\@gls@checkescactual{\@gls@checkescactual#3\null}%
    \fi
\else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
    \@gls@quotechar\string"\@gls@actualchar}%
    \ifx\null#3\null
        \def\@gls@checkescactual{\@gls@checkescactual#2\?\?\null}%
    \else
        \def\@gls@checkescactual{\@gls@checkescactual#2\?#3\null}%
    \fi
\fi
\@gls@checkescactual
}
\def\@gls@checkescbar#1\|#2\|#3\null{%
    \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
    \toks@={#1}%
    \ifx\null#2\null
        \ifx\null#3\null
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
            \def\@gls@checkescbar{\relax}%
        \else
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
            \@gls@quotechar\string"\@gls@encapchar
            \@gls@quotechar\string"\@gls@encapchar}%
            \def\@gls@checkescbar{\@gls@checkescbar#3\null}%
        \fi
    \else
        \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\string"\@gls@encapchar}%
        \ifx\null#3\null
            \def\@gls@checkescbar{\@gls@checkescbar#2\|\|\null}%
        \else
            \def\@gls@checkescbar{\@gls@checkescbar#2\|#3\null}%
        \fi
    \fi
\@gls@checkescbar
}
\def\@gls@checkesclevel#1\!#2\!#3\null{%
    \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
    \toks@={#1}%
    \ifx\null#2\null
        \ifx\null#3\null
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
            \def\@gls@checkesclevel{\relax}%
        \else
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
            \@gls@quotechar\string"\@gls@levelchar
            \@gls@quotechar\string"\@gls@levelchar}%
            \def\@gls@checkesclevel{\@gls@checkesclevel#3\null}%
        \fi
    \fi
}

```

```

\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!\!\null}%
\else
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!#3\null}%
\fi
\fi
\fi
\@gls@checkesclevel
}
\def\@gls@checkbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@gls@checkbar{\@gls@checkbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkbar{\@gls@checkbar#2|\!\null}%
\else
\def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}
\def\@gls@checklevel#1!#2!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checklevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
\def\@gls@checklevel{\@gls@checklevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar}%

```

```

\ifx\null#3\null
  \def\@gls@checklevel{\@gls@checklevel#2!\null}%
\else
  \def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
\fi
\fi
\@gls@checklevel
}
\def\@gls@checkactual#1?#2?#3\null{%
  \gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkactual{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
      \def\@gls@checkactual{\@gls@checkactual#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@gls@actualchar}%
    \ifx\null#3\null
      \def\@gls@checkactual{\@gls@checkactual#2??\null}%
    \else
      \def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
    \fi
  \fi
  \@gls@checkactual
}
\bgroup
\catcode'\0\relax
\catcode'\12\relax
\gdef\@gls@checkesc#1\#2\#3\null{%
  \gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkesc{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@backslashchar\@gls@quotechar\@backslashchar}%
      \def\@gls@checkesc{\@gls@checkesc#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@backslashchar}%
  \ifx\null#3\null

```

```

        |def|@gls@checkesc{|@gls@checkesc#2\\|null}%
    |else
        |def|@gls@checkesc{|@gls@checkesc#2\#3|null}%
    |fi
|fi
|@gls@checkesc
}
|egroup
\def\@gls@xdycheckquote#1"#2"#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@xdycheckquote{\relax}%
    |else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \string"\string"}%
      \def\@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
    |fi
  |else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \string"}%
    \ifx\null#3\null
      \def\@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
    |else
      \def\@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
    |fi
  |fi
  \@gls@xdycheckquote
}
\edef\def@gls@xdycheckbackslash{%
\noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
  ##2\@backslashchar##3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter
  {\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={##1}%
\noexpand\ifx\noexpand\null##2\noexpand\null
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{\relax}%
\noexpand|else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
  \@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
  \noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
\noexpand|fi
\noexpand|else

```

```

\newcommand\gls@checkedmkidx{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
  \@backslashchar\@backslashchar}%
\newcommand\gls@xdycheckbackslash{%
  \noexpand\def\noexpand\@gls@xdycheckbackslash{%
    \noexpand\@gls@xdycheckbackslash##2\@backslashchar
    \@backslashchar\noexpand\null}%
  \noexpand\else
  \noexpand\def\noexpand\@gls@xdycheckbackslash{%
    \noexpand\@gls@xdycheckbackslash##2\@backslashchar
    ##3\noexpand\null}%
  \noexpand\fi
  \noexpand\fi
  \noexpand\@gls@xdycheckbackslash
}%
}
\def\gls@xdycheckbackslash

\newlength\gls@tmplen
\newcommand*\glsdohypertarget}[2]{%
  \gls@showtarget{#1}%
  \settoheight{\gls@tmplen}{#2}%
  \raisebox{\gls@tmplen}{\hypertarget{#1}{}}#2%
}
\newcommand*\glsdohyperlink}[2]{%
  \gls@showtarget{#1}%
  \hyperlink{#1}{#2}%
}
\newcommand*\glsdonohyperlink}[2]{#2}
\ifcsundef{hyperlink}%
{%
  \let\@glslink\glsdonohyperlink
}%
{%
  \let\@glslink\glsdohyperlink
}
\ifcsundef{hypertarget}%
{%
  \let\@glstarget\@secondoftwo
}%
{%
  \let\@glstarget\glsdohypertarget
}
\newcommand*\glsdisablehyper{%
  \KV@glslink@hyperfalse
  \let\@glslink\glsdonohyperlink
  \let\@glstarget\@secondoftwo
}
\newcommand*\glsenablehyper{%
  \KV@glslink@hypertrue

```

```

\let\@glslink\glsdohyperlink
\let\@glstarget\glsdohypertarget
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}
\newcommand*{\@gls}[2][{}]{%
  \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2}[]]}%
}
\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@firstofthree
    \let\gls custom text\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
\newcommand*{\@Gls}[2][{}]{%
  \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2}[]]}%
}
\def\@Gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@secondofthree
    \let\gls custom text\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}

```



```

\newcommand*{\@GLS}[2] [] {%
  \new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2} []}%
}
\def\@GLS@#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@thirdofthree
    \let\gls custom text\@empty
    \def\gls insert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@gls link@local
      \gls local unset{#2}%
    \else
      \gls unset{#2}%
    \fi
  }%
  \gls post link hook
}
\newrobustcmd*{\glspl}{\@gls@hyp@opt\@glspl}
\newcommand*{\@glspl}[2] [] {%
  \new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2} []}%
}
\def\@glspl@#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\gls caps case\@firstofthree
    \let\gls custom text\@empty
    \def\gls insert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@gls link@local
      \gls local unset{#2}%
    \else
      \gls unset{#2}%
    \fi
  }%
  \gls post link hook
}
\newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
\newcommand*{\@Glspl}[2] [] {%
  \new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2} []}%
}
\def\@Glspl@#1#2[#3] {%
  \glsdoifexists{#2}%
  {%

```

```

\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@firstoftwo
\let\glsifcaps\@secondofthree
\let\glsifcustomtext\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\ifKV@glslink@local
  \glslocalunset{#2}%
\else
  \glsunset{#2}%
\fi
}%
\glspostlinkhook
}
\newrobustcmd*{\GLSpl}{\@gls@hyp@opt\@GLSpl}
\newcommand*{\@GLSpl}[2][]{%
  \new@ifnextchar[{\@GLSpl@{#1}{#2}}{\@GLSpl@{#1}{#2}[]}%
}
\def\@GLSpl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glsifcaps\@thirdofthree
    \let\glsifcustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\glsdisp}{\@gls@hyp@opt\@glsdisp}
\newcommand*{\@glsdisp}[3][]{%
  \glsdoifexists{#2}{%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glsifcaps\@firstofthree
    \def\glsifcustomtext{#3}%
    \def\glsinsert{}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else

```

```

        \glsunset{#2}%
    \fi
}%
\glspostlinkhook
}
\newcommand*{\@gls@link@nocheckfirsthyper}{}
\newcommand{\@gls@field@link}[3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \@gls@link{#1}{#2}{#3}%
    }%
    \glspostlinkhook
}
\newrobustcmd*{\glsstext}{\@gls@hyp@opt\@glsstext}
\newcommand*{\@glsstext}[2][{}]{%
    \new@ifnextchar[{\@glsstext@{#1}{#2}}{\@glsstext@{#1}{#2}[]]}
\def\@glsstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentrytext{#2}#3}%
}
\newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
\newcommand*{\@GLStext}[2][{}]{%
    \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2}[]]}
\def\@GLStext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrytext{#2}#3}}%
}
\newrobustcmd*{\Glsstext}{\@gls@hyp@opt\@Glsstext}
\newcommand*{\@Glsstext}[2][{}]{%
    \new@ifnextchar[{\@Glsstext@{#1}{#2}}{\@Glsstext@{#1}{#2}[]]}
\def\@Glsstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentrytext{#2}#3}%
}
\newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
\newcommand*{\@glsfirst}[2][{}]{%
    \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2}[]]}
\def\@glsfirst@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
}
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
\newcommand*{\@Glsfirst}[2][{}]{%
    \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2}[]]}
\def\@Glsfirst@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
\newcommand*{\@GLSfirst}[2][{}]{%
    \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2}[]]}
\def\@GLSfirst@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirst{#2}#3}}%
}
}

```

```

\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
\newcommand*{\@glsplural}[2] []{%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} []]}
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\newcommand*{\@Glsplural}[2] []{%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} []]}
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
\newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
\newcommand*{\@GLSplural}[2] []{%
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} []]}
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryplural{#2}#3}}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{\@glsfirstplural}[2] []{%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} []]}
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
\newcommand*{\@Glsfirstplural}[2] []{%
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2} []]}
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
\newcommand*{\@GLSfirstplural}[2] []{%
  \new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} []]}
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirstplural{#2}#3}}%
}
\newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
\newcommand*{\@glsname}[2] []{%
  \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2} []]}
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryname{#2}#3}%
}
\newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}
\newcommand*{\@Glsname}[2] []{%
  \new@ifnextchar[{\@Glsname@{#1}{#2}}{\@Glsname@{#1}{#2} []]}
\def\@Glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryname{#2}#3}%
}
\newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}
\newcommand*{\@GLSname}[2] []{%

```

```

    \new@ifnextchar [{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}]
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryname{#2}#3}}%
}
\newrobustcmd*{\glsdesc}{\@gls@hyp@opt\@glsdesc}
\newcommand*{\@glsdesc}[2] []{%
  \new@ifnextchar [{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2} []}]
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydesc{#2}#3}%
}
\newrobustcmd*{\Glsdesc}{\@gls@hyp@opt\@Glsdesc}
\newcommand*{\@Glsdesc}[2] []{%
  \new@ifnextchar [{\@Glsdesc@{#1}{#2}}{\@Glsdesc@{#1}{#2} []}]
\def\@Glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%
}
\newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\newcommand*{\@GLSdesc}[2] []{%
  \new@ifnextchar [{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}]
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydesc{#2}#3}}%
}
\newrobustcmd*{\glsdescplural}{\@gls@hyp@opt\@glsdescplural}
\newcommand*{\@glsdescplural}[2] []{%
  \new@ifnextchar [{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}]
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\Glsdescplural}{\@gls@hyp@opt\@Glsdescplural}
\newcommand*{\@Glsdescplural}[2] []{%
  \new@ifnextchar [{\@Glsdescplural@{#1}{#2}}{\@Glsdescplural@{#1}{#2} []}]
\def\@Glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\newcommand*{\@GLSdescplural}[2] []{%
  \new@ifnextchar [{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}]
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydescplural{#2}#3}}%
}
\newrobustcmd*{\glssymbol}{\@gls@hyp@opt\@glssymbol}
\newcommand*{\@glssymbol}[2] []{%
  \new@ifnextchar [{\@glssymbol@{#1}{#2}}{\@glssymbol@{#1}{#2} []}]
\def\@glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\Glsymbol}{\@gls@hyp@opt\@Glsymbol}
\newcommand*{\@Glsymbol}[2] []{%
  \new@ifnextchar [{\@Glsymbol@{#1}{#2}}{\@Glsymbol@{#1}{#2} []}]
\def\@Glsymbol@#1#2[#3]{%

```

```

    \@gls@field@link{#1}{#2}{\Glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\GLSsymbol}{\@gls@hyp@opt\@GLSsymbol}
\newcommand*{\@GLSsymbol}[2] [] {%
    \new@ifnextchar [ {\@GLSsymbol@{#1}{#2}} {\@GLSsymbol@{#1}{#2} [] }
}
\def \@GLSsymbol@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrysymbol{#2}#3}}%
}
\newrobustcmd*{\glssymbolplural}{\@gls@hyp@opt\@glssymbolplural}
\newcommand*{\@glssymbolplural}[2] [] {%
    \new@ifnextchar [ {\@glssymbolplural@{#1}{#2}} {\@glssymbolplural@{#1}{#2} [] }
}
\def \@glssymbolplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\Glsymbolplural}{\@gls@hyp@opt\@Glsymbolplural}
\newcommand*{\@Glsymbolplural}[2] [] {%
    \new@ifnextchar [ {\@Glsymbolplural@{#1}{#2}} {\@Glsymbolplural@{#1}{#2} [] }
}
\def \@Glsymbolplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
\newcommand*{\@GLSsymbolplural}[2] [] {%
    \new@ifnextchar [ {\@GLSsymbolplural@{#1}{#2}} {\@GLSsymbolplural@{#1}{#2} [] }
}
\def \@GLSsymbolplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrysymbolplural{#2}#3}}%
}
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
\newcommand*{\@glsuseri}[2] [] {%
    \new@ifnextchar [ {\@glsuseri@{#1}{#2}} {\@glsuseri@{#1}{#2} [] }
}
\def \@glsuseri@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
\newcommand*{\@Glsuseri}[2] [] {%
    \new@ifnextchar [ {\@Glsuseri@{#1}{#2}} {\@Glsuseri@{#1}{#2} [] }
}
\def \@Glsuseri@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\newcommand*{\@GLSuseri}[2] [] {%
    \new@ifnextchar [ {\@GLSuseri@{#1}{#2}} {\@GLSuseri@{#1}{#2} [] }
}
\def \@GLSuseri@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseri{#2}#3}}%
}
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
\newcommand*{\@glsuserii}[2] [] {%
    \new@ifnextchar [ {\@glsuserii@{#1}{#2}} {\@glsuserii@{#1}{#2} [] }
}
\def \@glsuserii@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
}

```

```

\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
\newcommand*{\@Glsuserii}[2] [] {%
  \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2} []}]
\def\@Glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\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*{\Glsuseriiii}{\@gls@hyp@opt\@Glsuseriiii}
\newcommand*{\@Glsuseriiii}[2] [] {%
  \new@ifnextchar[{\@Glsuseriiii@{#1}{#2}}{\@Glsuseriiii@{#1}{#2} []}]
\def\@Glsuseriiii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseriiii{#2}#3}%
}
\newrobustcmd*{\Glsuseriiiii}{\@gls@hyp@opt\@Glsuseriiiii}
\newcommand*{\@Glsuseriiiii}[2] [] {%
  \new@ifnextchar[{\@Glsuseriiiii@{#1}{#2}}{\@Glsuseriiiii@{#1}{#2} []}]
\def\@Glsuseriiiii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseriiiii{#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*{\Glsuserivv}{\@gls@hyp@opt\@Glsuserivv}
\newcommand*{\@Glsuserivv}[2] [] {%
  \new@ifnextchar[{\@Glsuserivv@{#1}{#2}}{\@Glsuserivv@{#1}{#2} []}]
\def\@Glsuserivv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserivv{#2}#3}%
}
\newrobustcmd*{\Glsuserivvi}{\@gls@hyp@opt\@Glsuserivvi}
\newcommand*{\@Glsuserivvi}[2] [] {%
  \new@ifnextchar[{\@Glsuserivvi@{#1}{#2}}{\@Glsuserivvi@{#1}{#2} []}]
\def\@Glsuserivvi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuserivvi{#2}#3}}%
}
\newrobustcmd*{\Glsuserivvii}{\@gls@hyp@opt\@Glsuserivvii}
\newcommand*{\@Glsuserivvii}[2] [] {%

```

```

\new@ifnextchar [{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2} []}]
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\Glsuserv}{\@gls@hyp@opt\@Glsuserv}
\newcommand*{\@Glsuserv}[2] []{%
\new@ifnextchar [{\@Glsuserv@{#1}{#2}}{\@Glsuserv@{#1}{#2} []}]
\def\@Glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserv{#2}#3}%
}
\newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
\newcommand*{\@GLSuserv}[2] []{%
\new@ifnextchar [{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2} []}]
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuserv{#2}#3}}%
}
\newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
\newcommand*{\@glsuservi}[2] []{%
  \new@ifnextchar [{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2} []}]
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
\newrobustcmd*{\Glsuservi}{\@gls@hyp@opt\@Glsuservi}
\newcommand*{\@Glsuservi}[2] []{%
  \new@ifnextchar [{\@Glsuservi@{#1}{#2}}{\@Glsuservi@{#1}{#2} []}]
\def\@Glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuservi{#2}#3}%
}
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\@ns@acrshort}
\newcommand*{\@ns@acrshort}[2] []{%
  \new@ifnextchar [{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2} []}]
}
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@secondoftwo
    \let\gls@scaps\@firstofthree
    \let\glsinsert\@empty
    \def\gls@customtext{%
      \acronymfont{\glsentryshort{#2}#3}%
    }%
    \@gls@link{#1}{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
}

```



```

\glspostlinkhook
}
\newrobustcmd*{\Acrshort}{\@gls@hyp@opt\ns@Acrshort}
\newcommand*{\ns@Acrshort}[2] [] {%
  \new@ifnextchar[{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2} []}]%
}
\def\@Acrshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\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] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\ns@ACRshortpl}
\newcommand*{\ns@ACRshortpl}[2] [] {%
  \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2} []}%
}
\def\@ACRshortpl#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\glentryshortpl{#2}}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrlong}{\@gls@hyp@opt\ns@acrlong}
\newcommand*{\ns@acrlong}[2] [] {%
  \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2} []}%
}

```

```

}
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsentrylong{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls@type @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrlong}{\@gls@hyp@opt\@ns@Acrlong}
\newcommand*{\ns@Acrlong}[2][]{%
  \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2}}[]}%
}
\def\@Acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylong{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls@type @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRlong}{\@gls@hyp@opt\@ns@ACRlong}
\newcommand*{\ns@ACRlong}[2][]{%
  \new@ifnextchar[{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2}}[]}%
}
\def\@ACRlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsapscase\@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\glsescapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\AcrLongpl}{\@gls@hyp@opt\ns@AcrLongpl}
\newcommand*{\ns@AcrLongpl}[2][]{%
  \new@ifnextchar[{\@AcrLongpl{#1}{#2}}{\@AcrLongpl{#1}{#2}[]}%
}
\def\@AcrLongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glsescapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRLongpl}{\@gls@hyp@opt\ns@ACRLongpl}
\newcommand*{\ns@ACRLongpl}[2][]{%
  \new@ifnextchar[{\@ACRLongpl{#1}{#2}}{\@ACRLongpl{#1}{#2}[]}%
}
\def\@ACRLongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

```

```

\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glsifcaps\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \mfirstucMakeUppercase{\glsentrylongpl{#2}#3}%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newcommand*{\@gls@entry@field}[2]{%
  \csname glo@\glsdetoklabel{#1}@#2\endcsname
}
\newcommand*{\glsletentryfield}[3]{%
  \letcs{#1}{glo@\glsdetoklabel{#2}@#3}%
}
\newcommand*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@#2}%
    \ifdef\@glo@text
    {%
      \xmakefirstuc{\@glo@text}%
    }%
    {%
      ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
        entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
        label and the field name}%
    }%
  }%
  {%
    ???%
  }%
}
\newcommand*{\glsentryname}[1]{\@gls@entry@field{#1}{name}}
\newrobustcmd*{\Glsentryname}[1]{%
  \@Gls@entryname{#1}%
}
\newcommand*{\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
\newcommand*{\@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@name}%
    \expandafter\@gls@getbody\@glo@text{}\@nil
    \expandafter\ifx\@gls@body\glsentrylong\relax
    \expandafter\Glsentrylong\@gls@rest
  }%
  \else

```

```

\expandafter\ifx\@gls@body\glsentryshort\relax
\expandafter\Glsentryshort\@gls@rest
\else
\expandafter\ifx\@gls@body\acronymfont\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\xmakefirstuc{\@glo@text}%
\fi
\fi
\fi
\fi
\fi
}%
{%
\@Gls@entry@field{#1}{name}%
}%
}
\newcommand*\glsentrydesc}[1]{\@Gls@entry@field{#1}{desc}}
\newrobustcmd*\Glsentrydesc}[1]{%
\@Gls@entry@field{#1}{desc}%
}
\newcommand*\glsentrydescplural}[1]{%
\@Gls@entry@field{#1}{descplural}%
}
\newrobustcmd*\Glsentrydescplural}[1]{%
\@Gls@entry@field{#1}{descplural}%
}
\newcommand*\glsentrytext}[1]{\@Gls@entry@field{#1}{text}}
\newrobustcmd*\Glsentrytext}[1]{%
\@Gls@entry@field{#1}{text}%
}
\newcommand*\glsentryplural}[1]{%
\@Gls@entry@field{#1}{plural}%
}
\newrobustcmd*\Glsentryplural}[1]{%
\@Gls@entry@field{#1}{plural}%
}
\newcommand*\glsentrysymbol}[1]{%
\@Gls@entry@field{#1}{symbol}%
}
\newrobustcmd*\Glsentrysymbol}[1]{%
\@Gls@entry@field{#1}{symbol}%
}

```

```

}
\newcommand*\glsentrysymbolplural}[1]{%
  \@gls@entry@field{#1}{symbolplural}%
}
\newrobustcmd*\Glsentrysymbolplural}[1]{%
  \@Gls@entry@field{#1}{symbolplural}%
}
\newcommand*\glsentryfirst}[1]{%
  \@gls@entry@field{#1}{first}%
}
\newrobustcmd*\Glsentryfirst}[1]{%
  \@Gls@entry@field{#1}{first}%
}
\newcommand*\glsentryfirstplural}[1]{%
  \@gls@entry@field{#1}{firstpl}%
}
\newrobustcmd*\Glsentryfirstplural}[1]{%
  \@Gls@entry@field{#1}{firstpl}%
}
\newcommand*\glscapitalisewords}[1]{%
  \capitalisewords{#1}%
}
\newrobustcmd*\@glsentrytitlecase}[2]{%
  \glsdoifexists{#1}%
  {%
    \glsfieldfetch{#1}{#2}{\@gls@value}%
    \expandafter\glscapitalisewords\expandafter{\@gls@value}%
  }%
}
\ifdef\texorpdfstring
{
  \newcommand*\glsentrytitlecase}[2]{%
    \texorpdfstring
    {\@glsentrytitlecase{#1}{#2}}%
    {\@gls@entry@field{#1}{#2}}%
  }
}
{
  \newcommand*\glsentrytitlecase}[2]{\@glsentrytitlecase{#1}{#2}}
}
\newcommand*\glsentrytype}[1]{\@gls@entry@field{#1}{type}}
\newcommand*\glsentrysort}[1]{%
  \@gls@entry@field{#1}{sort}%
}
\newcommand*\glsentryparent}[1]{%
  \@gls@entry@field{#1}{parent}%
}
\newcommand*\glsentryuseri}[1]{%
  \@gls@entry@field{#1}{useri}%
}
}

```

```

\newrobustcmd*\Glsentryuseri}[1]{%
  \@Gls@entry@field{#1}{useri}%
}
\newcommand*\glsentryuserii}[1]{%
  \@gls@entry@field{#1}{userii}%
}
\newrobustcmd*\Glsentryuserii}[1]{%
  \@Gls@entry@field{#1}{userii}%
}
\newcommand*\glsentryuseriii}[1]{%
  \@gls@entry@field{#1}{useriii}%
}
\newrobustcmd*\Glsentryuseriii}[1]{%
  \@Gls@entry@field{#1}{useriii}%
}
\newcommand*\glsentryuseriv}[1]{%
  \@gls@entry@field{#1}{useriv}%
}
\newrobustcmd*\Glsentryuseriv}[1]{%
  \@Gls@entry@field{#1}{useriv}%
}
\newcommand*\glsentryuserv}[1]{%
  \@gls@entry@field{#1}{userv}%
}
\newrobustcmd*\Glsentryuserv}[1]{%
  \@Gls@entry@field{#1}{userv}%
}
\newcommand*\glsentryuservi}[1]{%
  \@gls@entry@field{#1}{uservi}%
}
\newrobustcmd*\Glsentryuservi}[1]{%
  \@Gls@entry@field{#1}{uservi}%
}
}
\newcommand*\glsentryshort}[1]{\@gls@entry@field{#1}{short}}
\newrobustcmd*\Glsentryshort}[1]{%
  \@Gls@entry@field{#1}{short}%
}
}
\newcommand*\glsentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}
\newrobustcmd*\Glsentryshortpl}[1]{%
  \@Gls@entry@field{#1}{shortpl}%
}
}
\newcommand*\glsentrylong}[1]{\@gls@entry@field{#1}{long}}
\newrobustcmd*\Glsentrylong}[1]{%
  \@Gls@entry@field{#1}{long}%
}
}
\newcommand*\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}
\newrobustcmd*\Glsentrylongpl}[1]{%
  \@Gls@entry@field{#1}{longpl}%
}
}
\newcommand*\glsentryfull}[1]{%

```



```

\acrfullformat{\glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
}
\newrobustcmd*{\Glsentryfull}[1]{%
\acrfullformat{\Glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
}
\newcommand*{\glsentryfullpl}[1]{%
\acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newrobustcmd*{\Glsentryfullpl}[1]{%
\acrfullformat{\Glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newcommand*{\glsentrynumberlist}[1]{%
\glsdoifexists{#1}%
{%
\@gls@entry@field{#1}{numberlist}%
}%
}
\@ifpackageloaded{hyperref} {%
\newcommand*{\glsdisplaynumberlist}[1]{%
\GlossariesWarning
{%
\string\glsdisplaynumberlist\space
doesn't work with hyperref.^^JUsing
\string\glsentrynumberlist\space instead%
}%
\glsentrynumberlist{#1}%
}%
}%
{%
\newcommand*{\glsdisplaynumberlist}[1]{%
\glsdoifexists{#1}%
{%
\bgroup
\protected@edef\@glo@label{\glsdetoklabel{#1}}%
\let\@org@glsnumberformat\glsnumberformat
\def\glsnumberformat##1{##1}%
\protected@edef\the@numberlist{%
\csname glo@\@glo@label @numberlist\endcsname}%
\def\@gls@numlist@sep{}%
\def\@gls@numlist@nextsep{}%
\def\@gls@numlist@lastsep{}%
\def\@gls@thislist{}%
\def\@gls@donext@def{}%
\renewcommand\do[1]{%
\protected@edef\@gls@thislist{%
\@gls@thislist
\noexpand\@gls@numlist@sep
##1%
}%
\let\@gls@numlist@sep\@gls@numlist@nextsep

```

```

        \def\@gls@numlist@nextsep{\glsnumlistsep}%
        \@gls@donext@def
        \def\@gls@donext@def{%
            \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
        }%
    }%
}
\expandafter \glsnumlistparser \expandafter{\the@numberlist}%
\let\@gls@numlist@sep\@gls@numlist@lastsep
\@gls@thislist
\egroup
}%
}
}
\newcommand*{\glsnumlistsep}{, }
\newcommand*{\glsnumlistlastsep}{ \& }
\newcommand*{\gls hyperlink}[2][\glsentrytext{\@glo@label}]{%
    \def\@glo@label{#2}%
    \@glslink{\glolinkprefix\glsdetoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*{\glsadd}[2][{%
    \@gls@adjustmode
    \glsdoifexists{#2}%
    {%
        \def\@glsnumberformat{glsnumberformat}%
        \protected@edef\@gls@counter{\csname glo@\glsdetoklabel{#2}@counter\endcsname}%
        \setkeys{glossadd}{#1}%
        \@gls@saveentrycounter
        \@gls@setsort{#2}%
        \@do@wrglossary{#2}%
    }%
}
}
\newcommand*{\@gls@adjustmode}{}
\AtBeginDocument{\renewcommand*{\@gls@adjustmode}{\ifvmode\mbox{}\fi}}
\newrobustcmd*{\glsaddall}[1][{%
    \protected@edef\@glo@type{\@glo@types}%
    \setkeys{glossadd}{#1}%
    \forallglsentries[\@glo@type]{\@glo@entry}{%
        \glsadd[#1]{\@glo@entry}%
    }%
}
}
\newrobustcmd*{\glsaddallunused}[1][\@glo@types]{%
    \forallglsentries[#1]{\@glo@entry}%
    {%
        \ifglsused{\@glo@entry}{\glsadd[format=glsignore]{\@glo@entry}}%
    }%
}
}
\newcommand*{\glsignore}[1]{}
\edef\glsopenbrace{\expandafter\@gobble\string\{}}

```

```

\edef\glsclosebrace{\expandafter@gobble\string\}}
\edef\glsbackslash{\expandafter@gobble\string\}
\edef\glsquote#1{\string"#1\string"}
\edef\glspercentchar{\expandafter@gobble\string\%}
\edef\glstildechar{\string~}
\ifglsexindy
  \newcommand*{\@glsfirstletter}{A}
\fi
\newcommand*{\GlsSetXdyFirstLetterAfterDigits}{%
  \@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
\ifglsexindy
  \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*{\@glsfirstletter}{#1}}
  \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*{\@glsfirstletter}{#1}%
    \@onelevel@sanitize\@glsfirstletter
  }
\else
  \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \glsnoindywarning\GlsSetXdyFirstLetterAfterDigits}
  \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%
    \@GlsSetXdyFirstLetterAfterDigits
  }
\fi
\ifglsexindy
  \newcommand*{\@xdynumbergrouporder}{:before \string"@glsfirstletter\string"}
\fi
\newcommand*{\GlsSetXdyNumberGroupOrder}[1]{%
  \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
}
\ifglsexindy
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
  }
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
    \@onelevel@sanitize\@xdynumbergrouporder
  }
\else
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \glsnoindywarning\GlsSetXdyNumberGroupOrder}
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
    \@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*{\@glsminrange}{2}
\ifglsexindy
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \renewcommand*{\@glsminrange}{#1}}
\else
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%

```

```

\glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglsxindy
\def\writeist{%
\ifundef{\glswrite}{\newwrite\glswrite}{}%
\@gls@addpredefinedattributes
\openout\glswrite=\istfilename
\write\glswrite{;; xindy style file created by the glossaries
package}%
\write\glswrite{;; for document '\jobname' on
\the\year-\the\month-\the\day}%
\write\glswrite{^^J; required styles^^J}
\@for\@xdystyle:=\@xdyrequiredstyles\do{%
\ifx\@xdystyle\@empty
\else
\protected@write\glswrite{}{(require
\string"\@xdystyle.xdy\string")}%
\fi
}%
\write\glswrite{^^J%
; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%
\@for\@gls@classI:=\@gls@xdy@locationlist\do{%
\protected@write\glswrite{}{(define-location-class
\string"\@gls@classI\string"^^J\space\space\space
(
:sep "{{"
\csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
:sep "}"
)
^^J\space\space\space
:min-range-length \@glsminrange^^J%
)
}%
}%
\@for\@gls@classII:=\@gls@xdy@locationlist\do{%
\protected@write\glswrite{}{(define-location-class
\string"\@gls@classII-\@gls@classI\string"
^^J\space\space\space
(
:sep "{"
\csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
:sep "}"
\csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
:sep "}"
)
}
}
}

```

```

        ^^J\space\space\space
        :min-range-length \@glsminrange^^J%
    )
} %
} %
} %
} %
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
:unverified )}%
\write\glswrite{(markup-crossref-list
:class \string"see\string"^^J\space\space\space
:open \string"\string\glsseeformat\string"
:close \string"{}\string")}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
(\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
:open \string"\string
\glossarysection[\string\glossarytoctitle]{\string
\glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
{%
\@for\@this@attr:=\@xdyattributelist\do{%
\protected\write\glswrite{ }\string\providecommand*%
\expandafter\string
\csname glsX\@this@ctr X\@this@attr\endcsname[2]%
{%
\string\setentrycounter
[\expandafter\@gobble\string\#1]{\@this@ctr}%
\expandafter\string
\csname\@this@attr\endcsname
{\expandafter\@gobble\string\#2}%
}%
}%
}%
} %
} %
} %
\write\glswrite{%
\string\begin
{theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
\space\space:close \string"\glspercentchar\glstildechar n\string
\end{theglossary}\string\glossarypostamble
\glstildechar n\string" ^^J\space\space\space
:tree}%
\write\glswrite{(markup-letter-group-list

```

```

        :sep \string"\string\glsgroupskip\glstildechar n\string"}}%
\write\glswrite{(markup-indexentry
        :open \string"\string\relax \string\glsresetentrylist
        \glstildechar n\string"}}%
\write\glswrite{(markup-locclass-list :open
        \string"\glsopenbrace\string\glossaryentrynumbers
        \glsopenbrace\string\relax\space \string"^^J\space\space\space
        :sep \string", \string"
        :close \string"\glsclosebrace\glsclosebrace\string"}}%
\write\glswrite{(markup-locref-list
        :sep \string"\string\delimN\space\string"}}%
\write\glswrite{(markup-range
        :sep \string"\string\delimR\space\string"}}%
\@onelevel@sanitize\gls@suffiF
\@onelevel@sanitize\gls@suffiFF
\ifx\gls@suffiF\@empty
\else
        \write\glswrite{(markup-range
                :close "\gls@suffiF" :length 1 :ignore-end)}}%
\fi
\ifx\gls@suffiFF\@empty
\else
        \write\glswrite{(markup-range
                :close "\gls@suffiFF" :length 2 :ignore-end)}}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
        :sep \string"\string\glsgroupskip\glstildechar n\string"}}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
        :open-head \string"\string\glsgroupheading
        \glsopenbrace\string"^^J\space\space\space
        :close-head \string"\glsclosebrace\string"}}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}%
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax

```

```

\ifundef{\glswrite}{\newwrite\glswrite}{\relax
\openout\glswrite=\istfilename
\write\glswrite{\glspercentchar\space makeindex style file
  created by the glossaries package}
\write\glswrite{\glspercentchar\space for document
  'jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '@gls@actualchar'}
\write\glswrite{encap '@gls@encapchar'}
\write\glswrite{level '@gls@levelchar'}
\write\glswrite{quote '@gls@quotechar'}
\write\glswrite{keyword \string\string\glossaryentry\string}
\write\glswrite{preamble \string\string\glossarysection[\string
  \glossarytoctitle]{\string\glossarytitle}\string
  \glossarypreamble\string\n\string\begin{theglossary}\string
  \glossaryheader\string\n\string}
\write\glswrite{postamble \string\string%\string\n\string
  \end{theglossary}\string\glossarypostamble\string\n
  \string}
\write\glswrite{group_skip \string\string\glsgroupskip\string\n
  \string}
\write\glswrite{item_0 \string\string%\string\n\string}
\write\glswrite{item_1 \string\string%\string\n\string}
\write\glswrite{item_2 \string\string%\string\n\string}
\write\glswrite{item_01 \string\string%\string\n\string}
\write\glswrite{item_x1
  \string\string\relax \string\glsresetentrylist\string\n
  \string}
\write\glswrite{item_12 \string\string%\string\n\string}
\write\glswrite{item_x2
  \string\string\relax \string\glsresetentrylist\string\n
  \string}
\write\glswrite{delim_0 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_1 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_2 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_t \string\string\}\string}\string}
\write\glswrite{delim_n \string\string\delimN \string}
\write\glswrite{delim_r \string\string\delimR \string}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string\string\glsgroupheading\string\}\string}
\write\glswrite{heading_suffix
  \string\string}\string\relax
  \string\glsresetentrylist \string}
\write\glswrite{symhead_positive \string\glsymbols\string}
\write\glswrite{numhead_positive \string\glsnumbers\string}
\write\glswrite{page_compositor \string\glscompositor\string}
\@gls@escsdq\gls@suffixF

```

```

\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\fi
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{-}
\ifglsxindy
\newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\newcommand*\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}%
\ifpackageloaded{tracklang}%
{%
  \IfTrackedLanguage{german}%
  {%
    \def\@gls@extramakeindexopts{-g}%
  }%
  {}%
}%
{}%
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkquote####1####2####3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={####1}%
\noexpand\ifx\noexpand\null####2\noexpand\null
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%

```



```

\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###2#1#1\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote###2#1###3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\renewcommand\noexpand\@gls@checkmkidxchars}[1]{%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkquote###1\noexpand\@nil
#1#1\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkesquote###1\noexpand\@nil
\expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkesactual###1\noexpand\@nil
\noexpand\?\noexpand\?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkactual###1\noexpand\@nil
\noexpand?\noexpand?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkbar###1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkeschar###1\noexpand\@nil
\noexpand\|\noexpand\|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checklevel###1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null

```

```

\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkesclevel###1\noexpand\@nil
\noexpand\!\noexpand\!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkesc###1\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkescquote###1%
\expandonce{\csname#1\endcsname}###2\expandonce{\csname#1\endcsname}%
###3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={###1}%
\noexpand\ifx\noexpand\null###2\noexpand\null
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string
\expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###2\expandonce{\csname#1\endcsname}%
\expandonce{\csname#1\endcsname}\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###2\expandonce{\csname#1\endcsname}%
###3\noexpand\null}%
\noexpand\fi

```

```

        \noexpand\fi
        \noexpand\@gls@checkescquote
    }%
}
\@gls@docheckquotedef
}
\newcommand*\@gls@nosetquote}[1]{\PackageError{glossaries}%
  {\string\GlsSetQuote\space not permitted here}%
  {Move \string\GlsSetQuote\space earlier in the preamble, as
  soon as possible after glossaries.sty has been loaded}}
\fi
\newcommand*\@gls@extramakeindexopts}[1]{
\newcommand{\noist}{%
  \@gls@addpredefinedattributes
  \let\writeist\relax
}
\newcommand*\@makeglossary}[1]{%
  \ifglossaryexists{#1}%
  {%
    \ifglssavewrites
      \expandafter\newtoks\csname glo@#1@filetok\endcsname
    \else
      \expandafter\newwrite\csname glo@#1@file\endcsname
      \expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
    \fi
    \@gls@renewglossary
    \writeist
  }%
  {%
    \PackageError{glossaries}%
    {Glossary type ‘#1’ not defined}%
    {New glossaries must be defined before using \string\makeglossaries}%
  }%
}
\newcommand*\@glsopenfile}[2]{%
  \immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
  \PackageInfo{glossaries}{Writing glossary file
  \jobname.\csname @glotype@#2@out\endcsname}%
}
\newcommand*\@closegls}[1]{%
  \closeout\csname glo@#1@file\endcsname
}
\ifglxindy
\newcommand*\@gls@automake}[1]{%
  \ifglossaryexists{#1}
  {%
    \@closegls{#1}%
    \ifdefstring{\glsorder}{letter}%
    {\def\@gls@order{-M ord/letorder }}%
    {\let\@gls@order\@empty}%
  }

```

```

\ifcsundef{@xdy@#1@language}%
  {\let\@gls@langmod\@xdy@main@language}%
  {\letcs\@gls@langmod{@xdy@#1@language}}%
\edef\@gls@dothiswrite{\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \@gls@istfilebase\space
-C \@gls@codepage\space
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}%
}%
\@gls@dothiswrite
}%
{
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\else
\newcommand*{\@gls@automake}[1]{%
\ifglossaryexists{#1}
{%
\@closegls{#1}%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-l }}%
{\let\@gls@order\@empty}%
\edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
-s \istfilename\space
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}%
}%
\@gls@dothiswrite
}%
{
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\fi
\ifglsxindy
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%
\IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-M ord/letorder }}%
{\let\@gls@order\@empty}%
\ifcsundef{@xdy@#1@language}%

```

```

        {\let\@gls@langmod\@xdy@main@language}%
        {\letcs\@gls@langmod{\@xdy@#1@language}}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \gls@istfilebase\space
-C \gls@codepage\space
-t \jobname.\csuse{@gloptype@#1@log}
-o \jobname.\csuse{@gloptype@#1@in}
\jobname.\csuse{@gloptype@#1@out}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@gloptype@#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{@gloptype@#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{@gloptype@#1@log}
-o \jobname.\csuse{@gloptype@#1@in}
\jobname.\csuse{@gloptype@#1@out}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@gloptype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\fi
\newcommand*{\@warn@nomakeglossaries}{}
\newcommand*{\warn@nomakeglossaries}{\@warn@nomakeglossaries}
\newcommand{\@gls@automake@immediate}{%
\ifnum\gls@automake@nr=2\relax

```

```

\@for\@gls@type:=\@glo@types\do{%
  \ifdefempty{\@gls@type}{}%
  {\@gls@automake@immediate{\@gls@type}}%
}%
\glsautomakefalse
\renewcommand*{\@gls@doautomake}{}%
\fi
}
\newcommand*{\makeglossaries}{%
\@domakeglossaries
{%
  \@gls@@automake@immediate
  \ifundef{\glswrite}{\newwrite\glswrite}{}%
  \protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{}}
  \protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{}}
  \ifundef{\@gls@extramakeindexopts
  }{%
  {%
    \protected@write\@auxout{}{\string\providecommand
      \string\@gls@extramakeindexopts[1]{}}
    \protected@write\@auxout{}{\string\@gls@extramakeindexopts
      {\@gls@extramakeindexopts}}%
  }%
  \protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
  \protected@write\@auxout{}{\string\@glsorder{\glsorder}}
  \@for\@glo@type:=\@glo@types\do{%
    \ifthenelse{\equal{\@glo@type}{}}{ }{%
      \@makeglossary{\@glo@type}}%
    }%
  \renewcommand*\newglossary[4] [] {%
  \PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%
  \let\@makeglossary\@gobble
  \let\makeglossaries\relax
  \@disable@onlypremakeg
  \let\gls@checkseeallowed\relax
  \let\warn@nomakeglossaries\relax
  \let\warn@noprntglossary\@makeglossaries@warn@noprntglossary
  \ifgls@savenumberlist
    \edef\@gls@dodolistparser{\noexpand\DeclareListParser
      {\noexpand\glsnumlistparser}{\delimN}}%
    \@gls@dodolistparser
  \fi
  \let\makenoidxglossaries\@no@makeglossaries
  \renewcommand*{\@printgloss@setsort}{%
    \let\@glo@assign@sortkey\@glo@no@assign@sortkey
  }%
  \ifglsautomake

```

```

\renewcommand*{\@gls@doautomake}{%
  \for\@gls@type:=\@glo@types\do{%
    \ifdefempty{\@gls@type}{}%
    {\@gls@automake{\@gls@type}}%
  }%
}%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}
\newcommand\@makeglossaries@warn@noprintglossary{%
  \ifdefstring{\@glo@types}{,}%
  {%
    \GlossariesWarningNoLine{No glossaries have been defined}%
  }%
  {%
    \GlossariesWarningNoLine{No \string\printglossary\space
      or \string\printglossaries\space
      found. ^^J(Remove \string\makeglossaries\space if you
      don't want any glossaries.) ^^JThis document will not
      have a glossary}%
  }%
}%
\AtEndDocument{%
  \warn@nomakeglossaries
  \warn@noprintglossary
}
\newcommand*\makenoidxglossaries{%
  \@domakeglossaries
  {%
    \renewcommand{\@gls@noref@warn}[1]{%
      \GlossariesWarning{Empty glossary for
        \string\printnoidxglossary[type={##1}].
        Rerun may be required (or you may have forgotten to use
        commands like \string\gls)}%
    }%
    \let\@gls@checkmkidxchars\@gobble
    \glsesclocationsfalse
    \let\@do@@wrglossary\gls@noidxglossary
    \let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
    \let\gls@checkseeallowed\relax
    \renewcommand{\@do@seeglossary}[2]{%
      \protected@edef\@gls@label{\glsdetoklabel{##1}}%
      \protected@write\@auxout{}{%
        \string\@gls@reference
        {\csname glo@\@gls@label @type\endcsname}%
        {\@gls@label}}%
      {%
        \string\glsseeformat##2}%
    }%
  }%
}

```

```

    }%
  }%
} %
\AtBeginDocument
{%
  \write\@auxout{\string\providecommand\string\@gls@reference[3]{} }%
}%
\def\warn@noprintglossary{%
  \GlossariesWarningNoLine{No \string\printnoidxglossary\space
    or \string\printnoidxglossaries ^^J
    found. (Remove \string\makenoidxglossaries\space if you
    don't want any glossaries.)^^JThis document will not have a glossary}%
}%
\let\warn@nomakeglossaries\relax
\let\makeglossaries\@no@makeglossaries
\renewcommand*\@printgloss@setsort{%
  \let\@glo@assign@sortkey\@glo@assign@sortkey
  \def\@glo@sorttype{\@glo@default@sorttype}%
}%
\renewcommand*\new@glossaryentry[2]{%
  \PackageError{glossaries}{Glossary entries must be
    defined in the preamble^^Jwhen you use
    \string\makenoidxglossaries}%
  {Either move your definitions to the preamble or use
    \string\makeglossaries}%
}%
\renewcommand*\@gls@entrynumberlist[1]{%
  \letcs{\@gls@loclist}{glo@\@gls@detoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \glsnoidxloclist{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for '##1'. Either
        a rerun is required or you haven't referenced the entry}%
    }%
  }%
}%
\renewcommand*\@gls@displaynumberlist[1]{%
  \letcs{\@gls@loclist}{glo@\@gls@detoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \def\@gls@noidxloclist@sep{%
      \def\@gls@noidxloclist@sep{%
        \def\@gls@noidxloclist@sep{%
          \glsnumlistsep
        }%
        \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
      }%
    }%
  }%

```



```

    }%
  }%
  \def\@gls@noidxloclist@finalsep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
  \@gls@noidxloclist@finalsep
  \@gls@noidxloclist@prev
}%
{
  ??\glsdoifexists{##1}%
  {
    \GlossariesWarning{Missing location list for ‘##1’. Either
      a rerun is required or you haven’t referenced the entry}%
  }%
}%
}%
\renewcommand*\@glsnumberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo\@glsdetoklabel{##1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
  \let\@gls@org@glsseeformat\glsseeformat
  \let\glsnoidxdisplayloc##2\relax
  \let\glsseeformat##3\relax
  \ifdef\@gls@loclist
  {
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {
    ??\glsdoifexists{##1}%
    {
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
  \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
  \let\glsseeformat\@gls@org@glsseeformat
}%
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
\@glo@check@sortallowed\makenoidxglossaries
}%
}
\@onlypreamble{\makenoidxglossaries}
\newcommand*\@glsnumberlistloop}[2]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
\newcommand*\@glsnoidxnumberlistloophandler}[1]{%
  #1%
}

```

```

\newcommand*{\@no@makeglossaries}{%
  \PackageError{glossaries}{You can't use both
  \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
  together.}%
}
\newcommand{\@gls@nofwarn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
  is required to make \string\printnoidxglossary[type={#1}] work}%
}
\newcommand*{\@gls@noidxglossary}{%
  \protected@write\@auxout{%
    \string\@gls@reference
    {\csname glo@\@gls@label @type\endcsname}%
    {\@gls@label}%
    {\string\@glsnoidxdisplayloc
    {\@gls@counterprefix}%
    {\@gls@counter}%
    {\@glsnumberformat}%
    {\@glslocref}%
    }%
  }%
}
\providecommand\istfile{\glswrite}
\AtEndDocument{%
  \glswritefiles
}
\newcommand*{\@glswritefiles}{%
  \forallglossaries{\@gls@type}{%
    \ifcsundef{glo@\@gls@type @filetok}%
    {%
      \def\@gls@tmp{}%
    }%
    {%
      \edef\@gls@tmp{\expandafter\the
        \csname glo@\@gls@type @filetok\endcsname}%
    }%
    \ifx\@gls@tmp\@empty
      \ifx\@gls@type\@glsdefaulttype
        \GlossariesWarningNoLine{Glossary '\@gls@type' has no
          entries.^^JRemember to use package option 'nomain' if
you
          don't want to^^Juse the main glossary}%
      \else
        \GlossariesWarningNoLine{Glossary '\@gls@type' has no
          entries}%
      \fi
    \else
      \@glsopenfile{\glswrite}{\@gls@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{\theglsentrycounter}%
  \expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
  \ifx\@glsHlocref\@glslocref
    \def\@glo@counterprefix{}%
  \else
    \protected@edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
      {\@glslocref}{\@glsHlocref}%
    }%
    \@do@gls@getcounterprefix
  \fi
  \protected@edef\@gls@label{\glsdetoklabel{#1}}%
  \@do@wrglossary
}

```

```

\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodstrue
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
\begingroup
\let\gls@orgthe\the
\let\gls@orgnumber\number
\let\gls@orgarabic@arabic
\let\gls@orgromannumeral\romannumeral
\let\gls@orgalph\@alph
\let\gls@orgAlph\@Alph
\let\gls@orgRoman\@Roman
\ifglswrallowprimitivemods
\def\gls@the##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
\def\the{\expandafter\gls@the}%
\def\gls@number##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
\def\number{\expandafter\gls@number}%
\fi
\def\@arabic##1{%
\ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\romannumeral##1{%
\ifx##1\c@page \gls@romanpage\else\gls@orgromannumeral##1\fi}%
\def\@Roman##1{%
\ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
\ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
\ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\@glslocref{\theglsentrycounter}%
\endgroup
\@gls@checkmkidxchars\@glslocref
\expandafter\ifx\theglsentrycounter\theglsentrycounter\relax
\def\@glo@counterprefix{}%
\else
\protected@edef\@glsHlocref{\theglsentrycounter}%
\@gls@checkmkidxchars\@glsHlocref
\edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
{\@glslocref}{\@glsHlocref}%
}%
\@do@gls@getcounterprefix
\fi
\protected@edef\@gls@label{\glsdetoklabel{#1}}%
\@do@wrglossary
}
\newcommand*{\@do@wrglossary}{%
\ifglswrallowprimitivemods
\expandafter\@glo@checkmkidxrangechar\@glsnumberformat\@nil

```

```

\def\@glo@range{}%
\expandafter\if\@glo@prefix(\relax
\def\@glo@range{:open-range}%
\else
\expandafter\if\@glo@prefix)\relax
\def\@glo@range{:close-range}%
\fi
\fi
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
(indexentry :tkey (\csname glo@\@gls@label @index\endcsname)
:locref \string{\@glo@counterprefix}{\@gls@locref}\string" %
:attr \string"\@gls@counter\@glo@suffix\string"
\@glo@range
)
}%
\else
\@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@gls@numberformat}%
{\@glo@counterprefix}%
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
\string@glossaryentry{\csname glo@\@gls@label @index\endcsname
\@gls@encapchar\@glo@numfmt}{\@gls@locref}}%
\fi
}
\newcommand*\@gls@getcounterprefix[2]{%
\edef\@gls@thisloc{#1}\edef\@gls@thisHloc{#2}%
\ifx\@gls@thisloc\@gls@thisHloc
\def\@glo@counterprefix{}%
\else
\def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
\def\@glo@tmp{##2}%
\ifx\@glo@tmp\@empty
\def\@glo@counterprefix{}%
\else
\def\@glo@counterprefix{##1}%
\fi
}%
\@gls@get@counterprefix#2.#1\end@getprefix
\ifx\@glo@counterprefix\@empty
\GlossariesWarning{Hyper target ‘#2’ can’t be formed by
prefixing~Jlocation ‘#1’. You need to modify the
definition of \string\theH\@gls@counter~Jotherwise you
will get the warning: “name{\@gls@counter.#1}’ has been~J
referenced but does not exist”}%
\fi
\fi
}
\newcommand{\@do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref

```

```

\ifglxindy
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    (indexentry
      :tkey (\csname glo@#1@index\endcsname)
      :xref (\string"@gls@xref\string")
      :attr \string"see\string"
    )
  }%
\else
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    \string@glossaryentry{\csname glo@#1@index\endcsname
    \@gls@encapchar glsseeformat@gls@xref}{Z}}%
\fi
}
\def\@gls@fixbraces#1#2#3\@nil{%
  \ifx#2[\relax
    \@gls@fixbraces#1#2#3\@end@fixbraces
  \else
    \def#1{#{#2#3}}%
  \fi
}
\def\@gls@fixbraces#1[#2]#3\@end@fixbraces{%
  \def#1{[#2]{#3}}%
}
\newrobustcmd*{\glssee}[3][\seename]{%
  \@do@seeglossary{#2}{[#1]{#3}}}
\newcommand*{\@glssee}[3][\seename]{%
  \glssee[#1]{#3}{#2}}
\newrobustcmd*{\glsseeformat}[3][\seename]{%
  \emph{#1} \glsseelist{#2}}
\newrobustcmd*{\glsseelist}[1]{%
  \let\@gls@dolast\relax
  \let\@gls@donext\relax
  \@for\@gls@thislabel:=#1\do{%
    \ifx\@xfor@nextelement\@nnil
      \@gls@dolast
    \else
      \@gls@donext
    \fi
    \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
    \let\@gls@dolast\glsseelastsep
    \let\@gls@donext\glsseesep
  }%
}
\newcommand*{\glsseelastsep}{\space\andname\space}
\newcommand*{\glsseesep}{, }
\newrobustcmd*{\glsseeitem}[1]{\glshyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*{\glsseeitemformat}[1]{\glsentrytext{#1}}
\newcommand*{\gls@save@numberlist}[1]{%
  \ifglssavenumberlist

```

```

\toks@{#1}%
\protected@edef\do@writeaux@info{%
  \noexpand\csgdef{glo@glscurrententrylabel @numberlist}{\the\toks@}%
}%
\@onelevel@sanitize\do@writeaux@info
\protected@write\auxout{}{\do@writeaux@info}%
\fi
}
\newcommand*\warn@noprntglossary{}{}%
\ifcsundef{printglossary}{}{}%
{%
  \@gls@warnonglossdefined
  \undef\printglossary
}
\newcommand*\@printgloss@checkexists@noignored}[2]{%
  \ifglossaryexists{#1}%
  {#2}%
  {%
    \ifignoredglossary{#1}%
    {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
    {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
  }%
}
\newcommand*\@printgloss@checkexists@allowignored}[2]{%
  \s@ifglossaryexists{#1}%
  {#2}%
  {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\newcommand*\printglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@glossary}%
}
\newcommand*\printglossaries{%
  \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}
\newcommand*\printnoidxglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@noidx@glossary}%
}
\newcommand*\printnoidxglossaries{%
  \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
\newcommand*\@printgloss@setsort{}{}
\newcommand*\@gls@preglossaryhook{}{}
\newcommand*\printglossary}[2]{%
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{\csname @glo@type@\@glo@type @title\endcsname}%
  \def\glossarytoctitle{\glossarytitle}%
  \let\org@glossarytitle\glossarytitle

```



```

\def@glossarystyle{%
  \ifx@glossary@default@style\relax
    \GlossariesWarning{No default glossary style provided \MessageBreak
      for the glossary '@glo@type'. \MessageBreak
      Using deprecated fallback. \MessageBreak
      To fix this set the style with \MessageBreak
      \string\setglossarystyle\space or use the \MessageBreak
      style key=value option}%
  \fi
}%
\def@gls@dotoc@title{\gls@set@toc@title{\@glo@type}}%
\let@org@glossary@entry@numbers@glossary@entry@numbers
\bgroup
  \@print@gloss@set@sort
  \setkeys{print@gloss}{#1}%
  \@print@gloss@check@exists{\@glo@type}%
  {%
    \ifx@glossary@title@org@glossary@title
    \else
      \expandafter\let\csname @glo@type@\@glo@type @title\endcsname
      \glossary@title
    \fi
    \let\current@glossary@\@glo@type
    \let@org@glossary@entry@numbers@glossary@entry@numbers
    \let@gls@non@next@pages@gls@non@next@pages
    \let@gls@next@pages@gls@next@pages
    \let\nopost@desc@nopost@desc
    \gls@dotoc@title
    \@glossary@style
    \let@gls@org@glossary@entry@field@gloss@entry
    \let@gls@org@glossary@sub@entry@field@sub@gloss@entry
    \renewcommand{\gloss@entry}[1]{%
      \protected@xdef\gls@current@entry@label{\gls@detok@label{##1}}%
      \gls@org@glossary@entry@field{##1}%
    }%
    \renewcommand{\sub@gloss@entry}[2]{%
      \protected@xdef\gls@current@entry@label{\gls@detok@label{##2}}%
      \gls@org@glossary@sub@entry@field{##1}{##2}%
    }%
    \@gls@preglossary@hook
    #2%
  }%
\egroup
\global\let@glossary@entry@numbers@org@glossary@entry@numbers
\global\let\warn@noprint@glossary\relax
}
\newcommand{\@print@glossary}{%
  \makeatletter
  \@input@{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%
  \IfFileExists{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%

```

```

}%
{\null}%
\ifglxindy
  \ifcsundef{@xdy@\@glo@type @language}%
  {%
    \edef\@do@auxoutstuff{%
      \noexpand\AtEndDocument{%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\providecommand\string\@xdylanguage[2]{}%
        }%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\@xdylanguage{\@glo@type}{\@xdy@main@language}}%
        }%
      }%
    }%
  }%
  {%
    \edef\@do@auxoutstuff{%
      \noexpand\AtEndDocument{%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\providecommand\string\@xdylanguage[2]{}%
        }%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\@xdylanguage{\@glo@type}{\csname @xdy@\@glo@type
            @language\endcsname}}%
        }%
      }%
    }%
  }%
  \@do@auxoutstuff
  \edef\@do@auxoutstuff{%
    \noexpand\AtEndDocument{%
      \noexpand\immediate\noexpand\write\@auxout{%
        \string\providecommand\string\@gls@codepage[2]{}%
      }%
      \noexpand\immediate\noexpand\write\@auxout{%
        \string\@gls@codepage{\@glo@type}{\@gls@codepage}}%
      }%
    }%
  }%
  \@do@auxoutstuff
\fi
\renewcommand*{\@warn@nomakeglossaries}{%
  \GlossariesWarningNoLine{\string\makeglossaries\space
    hasn't been used,^^Jthe glossaries will not be updated}%
}%
}
\newcommand*{\@glo@sortentries}[2]{%
  \glosortentrieswarning
  \def\@glo@sortinglist{}%
  \def\@glo@sortinghandler{#1}%
  \protected@edef\@glo@type{#2}%
  \forlistcsloop{\@glo@do@sortentries}{\@glsref@#2}%
  \csdef{\@glsref@#2}{%
    \for\@this@label:=\@glo@sortinglist\do{%
      \xifinlistcs{\@this@label}{\@glsref@#2}%
    }%
  }%
}

```

```

    {}%
    {%
      \listcsxadd{@glsref@#2}{\@this@label}%
    }%
    \ifcsdef{@glo@sortingchildren@\@this@label}%
    {%
      \glo@addchildren{#2}{\@this@label}%
    }%
    {}%
  }%
}
\newcommand*{\@glo@addchildren}[2]{%
  \bgroup
  \letcs{\@glo@childlist}{@glo@sortingchildren@#2}%
  \@for\@this@childlabel:=\@glo@childlist\do
  {%
    \xifinlistcs{\@this@childlabel}{@glsref@#1}%
    {}%
    {%
      \listcsxadd{@glsref@#1}{\@this@childlabel}%
    }%
    \ifcsdef{@glo@sortingchildren@\@this@childlabel}%
    {%
      \glo@addchildren{#1}{\@this@childlabel}%
    }%
    {}%
    }%
  }%
\egroup
}
\newcommand*{\@glo@do@sortentries}[1]{%
  \ifglshasparent{#1}%
  {%
    \protected@edef\@glo@parent{\csuse{glo@glsdetoklabel{#1}@parent}}%
    \ifcsundef{@glo@sortingchildren@\@glo@parent}%
    {%
      \csdef{@glo@sortingchildren@\@glo@parent}{}%
    }%
    {}%
    \expandafter\@glo@sortedinsert
      \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
    \xifinlistcs{\@glo@parent}{@glsref@\@glo@type}%
    {}%
    }%
    {}%
    \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
  }%
  }%
  {%
    \@glo@sortedinsert{\@glo@sortinglist}{#1}%
  }%
}

```

```

    }%
  }
  \newcommand*{\@glo@sortedinsert}[2]{%
    \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
  }%
  \newcommand*{\@glo@sorthandler@word}[2]{%
    \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef\glo@do@compare{%
      \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
      {\expandonce\@gls@sort@B}%
      {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
  }
  \newcommand*{\@glo@sorthandler@letter}[2]{%
    \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef\glo@do@compare{%
      \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
      {\expandonce\@gls@sort@B}%
      {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
  }
  \newcommand*{\@glo@sorthandler@case}[2]{%
    \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef\glo@do@compare{%
      \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
      {\expandonce\@gls@sort@B}%
      {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
  }
  \newcommand*{\@glo@sorthandler@nocase}[2]{%
    \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef\glo@do@compare{%
      \noexpand\dtlicompare{\noexpand\dtl@sortresult}%
      {\expandonce\@gls@sort@B}%
      {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
  }
  \newcommand*{\@glo@sortmacro@word}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
      \@glo@sortentries{\@glo@sorthandler@word}{#1}%
    }%
  }%

```

```

    {%
      \PackageError{glossaries}{Conflicting sort options:^^J
        \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
        \string\printnoidxglossary[sort=word]}{}}%
    }%
  }
\newcommand*{\@glo@sortmacro@letter}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=letter]}{}}%
  }%
}
\newcommand*{\@glo@sortmacro@standard}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \ifcsdef{\@glo@sorthandler@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@glsdetoklabel{#1}@sort}%
\expandafter\glo@grabfirst\@gls@sort{}{} \@nil
\ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
{}%
{%
\ifdefempty{\@gls@currentlettergroup}{}%
{%
\global\let\@glo@thislettergrp\@glo@thislettergrp
\glsgroupskip
}%
\glsgroupheading{\@glo@thislettergrp}%
}%
\global\let\@gls@currentlettergroup\@glo@thislettergrp
\ifdefvoid{\@gls@loclist}
{%
\glossentry{#1}{}%
}%
{%
\glossentry{#1}%
{%
\glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
}%
}%
}%
}
\newcommand*{\glsnoidxloclist}[1]{%
\def\@gls@noidxloclist@sep{}%
\def\@gls@noidxloclist@prev{}%
\forlistloop{\glsnoidxloclisthandler}{#1}%
}
\newcommand*{\glsnoidxloclisthandler}[1]{%
\ifdefstring{\@gls@noidxloclist@prev}{#1}%
{%
}%
{%
\@gls@noidxloclist@sep
#1%
\def\@gls@noidxloclist@sep{\delimN}%
\def\@gls@noidxloclist@prev{#1}%
}%
}
\newcommand*{\glsnoidxdisplayloclisthandler}[1]{%
\ifdefstring{\@gls@noidxloclist@prev}{#1}%
{%
}%
{%
}

```



```

        \@gls@noidxloclist@sep
        \@gls@noidxloclist@prev
        \def\@gls@noidxloclist@prev{#1}%
    }%
}
\newcommand*\glsnoidxdisplayloc[4]{%
    \setentrycounter[#1]{#2}%
    \csuse{#3}{#4}%
}
\newcommand*\@gls@reference}[3]{%
    \glsdoifexistsorwarn{#2}%
    {%
        \ifcsundef{\@glsref@#1}{\csgdef{\@glsref@#1}{}}{%
            \ifinlistcs{#2}{\@glsref@#1}%
                {}%
            {\listcsgadd{\@glsref@#1}{#2}}%
            \ifcsundef{glo@\glsdetoklabel{#2}@loclist}%
                {\csgdef{glo@\glsdetoklabel{#2}@loclist}{}}%
                {}%
            \listcsgadd{glo@\glsdetoklabel{#2}@loclist}{#3}%
        }%
    }
}
\define@key{printgloss}{type}{\def\@glo@type{#1}}
\define@key{printgloss}{title}{%
    \def\glossarytitle{#1}%
    \let\gls@dotocitle\relax
}
\define@key{printgloss}{toctitle}{%
    \def\glossarytoctitle{#1}%
    \let\gls@dotocitle\relax
}
\define@key{printgloss}{style}{%
    \ifcsundef{\@glsstyle@#1}%
        {%
            \PackageError{glossaries}%
            {Glossary style ‘#1’ undefined}{}%
        }%
        {%
            \def\@glossarystyle{\setglossentrycompatibility
                \csname @glsstyle@#1\endcsname}%
        }%
    }
}
\define@choicekey{printgloss}{numberedsection}%
[\@gls@numberedsection@val\@gls@numberedsection@nr]%
{false,nolabel,autolabel,nameref}[nolabel]%
{%
    \ifcase\@gls@numberedsection@nr\relax
        \renewcommand*\@@glossarysecstar{*}%
        \renewcommand*\@@glossaryseclabel{}%
    \or

```

```

\renewcommand*\@@glossarysecstar}{}%
\renewcommand*\@@glossaryseclabel}{}%
\or
\renewcommand*\@@glossarysecstar}{}%
\renewcommand*\@@glossaryseclabel}{\label{\glsautoprefix\@glo@type}}%
\or
\renewcommand*\@@glossarysecstar}{*}%
\renewcommand*\@@glossaryseclabel}{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}}%
\fi
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
\csuse{glsnogroupskip#1}}%
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
\csuse{glsnopostdot#1}}%
}
\newcommand*\GlsEntryCounterLabelPrefix{glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
\csuse{glsentrycounter#1}}%
\@gls@define@glossaryentrycounter
}
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
\csuse{glsesubentrycounter#1}}%
\@gls@define@glossarysubentrycounter
}
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
\ifglsnonumberlist
\def\glossaryentrynumbers##1{}}%
\else
\def\glossaryentrynumbers##1{##1}}%
\fi}
\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
\newcommand*\@glo@no@assign@sortkey[1]{%
\PackageError{glossaries}{‘sort’ key not permitted with
\string\printglossary}%
{The ‘sort’ key may only be used with \string\printnoidxglossary}%
}
\newcommand*\@glo@assign@sortkey[1]{%
\def\@glo@sorttype{#1}}%
}
\newcommand*\@glsnonextpages}{%
\gdef\glossaryentrynumbers##1{%
\glsresetentrylist
}%
}
\newcommand*\@glsnextpages}{%
\gdef\glossaryentrynumbers##1{%
##1\glsresetentrylist}}

```

```

\newcommand*\glsresetentrylist}{%
  \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
\newcommand*\glsnonextpages}{}
\newcommand*\glsnextpages}{}
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
\newcommand*\glsresetsubentrycounter}{%
  \ifglssubentrycounter
    \setcounter{glossarysubentry}{0}%
  \fi
}
\newcommand*\glsresetentrycounter}{%
  \ifglentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*\glsstepentry}[1]{%
  \ifglentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
\newcommand*\glsstepsubentry}[1]{%
  \ifglssubentrycounter
    \protected@edef\currentglssubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
  \fi
}
\newcommand*\glsrefentry}[1]{%
  \ifglentrycounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else
    \ifglssubentrycounter
      \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
      \gls{#1}%
    \fi
  \fi
}
\newcommand*\glsentrycounterlabel}{%
  \ifglentrycounter
    \theglossaryentry.\space
  \fi
}
\newcommand*\glssubentrycounterlabel}{%
  \ifglssubentrycounter
    \theglossarysubentry)\space
  \fi
}

```

```

\newcommand*\glentryitem}[1]{%
  \ifglentrycounter
    \glstepentry{#1}\glentrycounterlabel
  \else
    \glresetsubentrycounter
  \fi
}
\newcommand*\glssubentryitem}[1]{%
  \ifglssubentrycounter
    \glstepsubentry{#1}\glssubentrycounterlabel
  \fi
}
\ifcsundef{theglossary}%
{%
  \newenvironment{theglossary}{}{}%
}%
{%
  \@gls@warnontheGLOSSDEFINED
  \renewenvironment{theglossary}{}{}%
}
\newcommand*\glossaryheader{}
\newcommand*\glstarget}[2]{\@glstarget{\glo@linkprefix#1}{#2}}
\providecommand*\compatibleglossentry}[2]{%
  \toks@{#2}%
  \protected@edef\@do@glossentry{\noexpand\glossaryentryfield{#1}%
    {\noexpand\glsnamefont
      {\expandafter\expandonce\csname glo@#1@name\endcsname}}%
    {\expandafter\expandonce\csname glo@#1@desc\endcsname}%
    {\expandafter\expandonce\csname glo@#1@symbol\endcsname}%
    {\the\toks@}}%
  }%
  \@do@glossentry
}
\newcommand*\glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
    \expandafter\glsnamefont\expandafter{\glo@name}%
  }%
}
\newcommand*\Glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsnamefont{\Glsentryname{#1}}%
  }%
}
\newcommand*\glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glentrydesc{#1}%
  }%
}

```

```

    }%
  }
  \newcommand*\Glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \Glsentrydesc{#1}%
    }%
  }
  \newcommand*\glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \glsentrysymbol{#1}%
    }%
  }
  \newcommand*\Glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \Glsentrysymbol{#1}%
    }%
  }
  \providecommand*\compatiblesubglossentry}[3]{%
    \toks@{#3}%
    \protected@edef\@do@subglossentry{\noexpand\glossarysubentryfield{\number#1}%
    {#2}}%
    {\noexpand\glsnamefont
     {\expandafter\expandonce\csname glo@#2@name\endcsname}}%
    {\expandafter\expandonce\csname glo@#2@desc\endcsname}%
    {\expandafter\expandonce\csname glo@#2@symbol\endcsname}%
    {\the\toks@}%
  }%
  \@do@subglossentry
}
\newcommand*\setglossentrycompatibility){%
  \let\glossentry\compatibleglossentry
  \let\subglossentry\compatiblesubglossentry
}
\setglossentrycompatibility
\newcommand*\glossaryentryfield}[5]{%
  \GlossariesWarning
  {Deprecated use of \string\glossaryentryfield.^.^J
   I recommend you change to \string\glossentry.^.^J
   If you've just upgraded, try removing your gls auxiliary
   files^^J and recompile}%
  \noindent\textbf{\glstarget{#1}{#2}} #4 #3. #5\par}
\newcommand*\glossarysubentryfield}[6]{%
  \GlossariesWarning
  {Deprecated use of \string\glossarysubentryfield.^.^J
   I recommend you change to \string\subglossentry.^.^J
   If you've just upgraded, try removing your gls auxiliary
   files^^J and recompile}%

```

```

\glstarget{#2}{\strut}#4. #6\par}
\newcommand*{\glsgroupskip}{}
\newcommand*{\glsgroupheading}[1]{}
\newcommand*{\glsgetgrouptitle}[1]{%
  \@gls@getgrouptitle{#1}{\@gls@grptitle}%
  \@gls@grptitle
}
\newcommand*{\@gls@getgrouptitle}[2]{%
  \dtl@ifsingle{#1}%
  {%
    \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
  }%
  {%
    \ifboolexpr{test{\ifstrequal{#1}{glsymbols}}
      or test{\ifstrequal{#1}{glsnumbers}}}%
    {%
      \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
    }%
    {%
      \def#2{#1}%
    }%
  }%
}
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%
  \DTLifint{#1}%
  {\edef#2{\char#1\relax}}%
  {%
    \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
  }%
}
\newcommand*{\glsgetgrouplabel}[1]{%
  \ifthenelse{\equal{#1}{\glsymbolsgroupname}}{glsymbols}{%
  \ifthenelse{\equal{#1}{\glsnumbersgroupname}}{glsnumbers}{#1}}%
\newcommand*{\setentrycounter}[2][1]{%
  \def\@glo@counterprefix{#1}%
  \ifx\@glo@counterprefix\@empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1}%
  \fi
  \def\glsentrycounter{#2}%
}
\newcommand*{\setglossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \csname @glsstyle@#1\endcsname
  }%
}

```

```

\ifx\@glossary@default@style\relax
  \protected@edef\@glossary@default@style{#1}%
\fi
}
\newcommand*\@glossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \GlossariesWarning
    {Deprecated command \string\glossarystyle.^^J
    I recommend you switch to \string\setglossarystyle\space unless
    you want to maintain backward compatibility}%
    \setglossentrycompatibility
    \csname @glsstyle@#1\endcsname
    \ifcsdef{@glscompstyle@#1}%
    {\setglossentrycompatibility\cuse{@glscompstyle@#1}}%
    {}%
  }%
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\newcommand*\@newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
\newcommand*\@renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
\newcommand*\@glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\glsnumber#1{#1}%
}%
{%
  \def\glsnumber#1{\@glsnumber#1\nohyperpage{}}\@nil}
}

```

```

\def\@glshypernumber#1\nohyperpage#2#3\@nil{%
  \ifx\#1\%
  \else
    \@delimR#1\delimR\delimR\%
  \fi
  \ifx\#2\%
  \else
    #2%
  \fi
  \ifx\#3\%
  \else
    \@glshypernumber#3\@nil
  \fi
}
\def\@delimR#1\delimR #2\delimR #3\{%
\ifx\#2\%
  \@delimN{#1}%
\else
  \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
\fi}
\def\@delimN#1{\@delimN#1\delimN \delimN\}
\def\@delimN#1\delimN #2\delimN#3\{%
\ifx\#3\%
  \@gls@numberlink{#1}%
\else
  \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
\fi
}
\def\@gls@numberlink#1{%
\begingroup
\toks@={}%
\@gls@removespaces#1 \@nil
\endgroup}
\def\@gls@removespaces#1 #2\@nil{%
\toks@=\expandafter{\the\toks@#1}%
\ifx\#2\%
  \edef\@gls@x{\the\toks@}%
  \ifx\@gls@x\empty
  \else
    \hyperlink{\glsentrycounter\@glo@counterprefix\the\toks@}%
      {\the\toks@}%
  \fi
\else
  \@gls@ReturnAfterFi{%
    \@gls@removespaces#2\@nil
  }%
\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
\newcommand*\hyperrm[1]{\textrm{\gls@hypernumber{#1}}}

```



```

\newcommand*\hypersf}[1]{\textsf{\glshypernumber{#1}}}
\newcommand*\hypertt}[1]{\texttt{\glshypernumber{#1}}}
\newcommand*\hyperbf}[1]{\textbf{\glshypernumber{#1}}}
\newcommand*\hypermd}[1]{\textmd{\glshypernumber{#1}}}
\newcommand*\hyperit}[1]{\textit{\glshypernumber{#1}}}
\newcommand*\hypersl}[1]{\textsl{\glshypernumber{#1}}}
\newcommand*\hyperup}[1]{\textup{\glshypernumber{#1}}}
\newcommand*\hypersc}[1]{\textsc{\glshypernumber{#1}}}
\newcommand*\hyperemph}[1]{\emph{\glshypernumber{#1}}}
\newcommand{\oldacronym}[4][\gls@label]{%
  \def\gls@label{#2}%
  \newacronym[#4]{#1}{#2}{#3}%
  \ifcsundef{xspace}%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}}%
    }%
  }%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
        \noexpand\gls{#1}\noexpand\xspace}%
      }%
    }%
  }%
}
\newcommand{\newacronym}[4][[]]{
\newcommand*\acrpluralsuffix}{\glsacrpluralsuffix}
\newrobustcmd*\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
\newcommand*\glsshortkey}{short}
\newcommand*\glsshortpluralkey}{shortplural}
\newcommand*\glslongkey}{long}
\newcommand*\glslongpluralkey}{longplural}
\newrobustcmd*\acrfull}{\@gls@hyp@opt\@ns@acrfull}
\newcommand*\ns@acrfull[2][[]]{%
  \new@ifnextchar[{\@acrfull{#1}{#2}}{%
    {\@acrfull{#1}{#2}[]}%
  }
}
\def\@acrfull#1#2[#3]{%
  \acrfullfmt{#1}{#2}{#3}%
}
\newcommand*\acrfullfmt}[3]{%
  \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newcommand{\acrlinkfullformat}[5]{%
  \acrfullformat{#1}{#3}{#4}{#5}{#2}{#3}{#4}[]}%
}
\newcommand{\acrfullformat}[2]{#1\glspace{#2}}
\newrobustcmd{\glspace}{\space}
\newrobustcmd*\Acrfull}{\@gls@hyp@opt\@ns@Acrfull}
\newcommand*\ns@Acrfull[2][[]]{%

```

```

\new@ifnextchar [{\@Acrfull{#1}{#2}}%
    {\@Acrfull{#1}{#2} []}%
}
\def\@Acrfull#1#2[#3]{%
    \Acrfullfmt{#1}{#2}{#3}%
}
\newcommand*\Acrfullfmt[3]{%
    \acrlinkfullformat{\@Acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newrobustcmd*\@ACRfull{\@gls@hyp@opt\ns@ACRfull}
\newcommand*\ns@ACRfull[2] []{%
    \new@ifnextchar [{\@ACRfull{#1}{#2}}%
        {\@ACRfull{#1}{#2} []}%
}
}
\def\@ACRfull#1#2[#3]{%
    \ACRfullfmt{#1}{#2}{#3}%
}
\newcommand*\ACRfullfmt[3]{%
    \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
\newrobustcmd*\@acrfullpl{\@gls@hyp@opt\ns@acrfullpl}
\newcommand*\ns@acrfullpl[2] []{%
    \new@ifnextchar [{\@acrfullpl{#1}{#2}}%
        {\@acrfullpl{#1}{#2} []}%
}
}
\def\@acrfullpl#1#2[#3]{%
    \acrfullplfmt{#1}{#2}{#3}%
}
\newcommand*\acrfullplfmt[3]{%
    \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*\@Acrfullpl{\@gls@hyp@opt\ns@Acrfullpl}
\newcommand*\ns@Acrfullpl[2] []{%
    \new@ifnextchar [{\@Acrfullpl{#1}{#2}}%
        {\@Acrfullpl{#1}{#2} []}%
}
}
\def\@Acrfullpl#1#2[#3]{%
    \ACRfullplfmt{#1}{#2}{#3}%
}
}

```

```

\newcommand*\ACRfullplfmt}[3]{%
  \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand*\acrnameformat}[2]{\acronymfont{#1}}
\newtoks\glskeylisttok
\newtoks\glslabeltok
\newtoks\glsshorttok
\newtoks\glslongtok
\newcommand*\newacronymhook{}
\newcommand*\SetGenericNewAcronym{%
  \let\@Gls@entryname\@Gls@acrentryname
  \renewcommand{\newacronym}[4][]{%
    \ifdefempty{\@glsacronymlists}%
    {%
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
    }%
    {}%
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\expandonce{\acronymentry{##2}}},%
      sort={\acronymssort{\the\glsshorttok}{\the\glslongtok}},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      \GenericAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}%
\renewcommand*\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}%
\renewcommand*\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}%
\renewcommand*\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}%

```

```

\renewcommand*\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}%
\renewcommand*\glsentryfull}[1]{\genacrfullformat{##1}{}}%
\renewcommand*\Glsentryfull}[1]{\Genacrfullformat{##1}{}}%
\renewcommand*\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}%
\renewcommand*\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}%
}
\newcommand*\GenericAcronymFields{description={\the\glslongtok}}
\newcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
\newcommand*\acronymsort}[2]{#1}
\newcommand*\setacronymstyle}[1]{%
  \ifcsundef{@glsacr@dispstyle@#1}
  {%
    \PackageError{glossaries}{Undefined acronym style ‘#1’}{%
  }%
  {%
    \ifdefempty{\@glsacronymlists}%
    {%
      \DeclareAcronymList{acronymtype}%
    }%
    {}%
    \SetGenericNewAcronym
    \GlsUseAcrStyleDefs{#1}%
    \@for\@gls@type:=\@glsacronymlists\do{%
      \defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{#1}}%
    }%
  }%
}
\newcommand*\newacronymstyle}[3]{%
  \ifcsdef{@glsacr@dispstyle@#1}%
  {%
    \PackageError{glossaries}{Acronym style ‘#1’ already exists}{%
  }%
  {%
    \csdef{@glsacr@dispstyle@#1}{#2}%
    \csdef{@glsacr@styledefs@#1}{#3}%
  }%
}
\newcommand*\renewacronymstyle}[3]{%
  \ifcsdef{@glsacr@dispstyle@#1}%
  {%
    \csdef{@glsacr@dispstyle@#1}{#2}%
    \csdef{@glsacr@styledefs@#1}{#3}%
  }%
  {%

```

```

    \PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{%
  }%
}
\newcommand*\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
{%
  \ifglshaslong{glslabel}{glsgenacfmt}{glsgenentryfmt}%
}%
{%
  \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
  \renewcommand*\genacrfullformat}[2]{%
    \glsentrylong{##1}##2\space
    (\protect\firstacronymfont{glsentryshort{##1}})%
  }%
  \renewcommand*\Genacrfullformat}[2]{%
    \Glsentrylong{##1}##2\space
    (\protect\firstacronymfont{glsentryshort{##1}})%
  }%
  \renewcommand*\genplacrfullformat}[2]{%
    \glsentrylongpl{##1}##2\space
    (\protect\firstacronymfont{glsentryshortpl{##1}})%
  }%
  \renewcommand*\Genplacrfullformat}[2]{%
    \Glsentrylongpl{##1}##2\space
    (\protect\firstacronymfont{glsentryshortpl{##1}})%
  }%
  \renewcommand*\acronymentry}[1]{\acronymfont{glsentryshort{##1}}}%
  \renewcommand*\acronymsort}[2]{##1}%
  \renewcommand*\acronymfont}[1]{##1}%
  \renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
  \renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
  \ifglshaslong{glslabel}{glsgenacfmt}{glsgenentryfmt}%
}%
{%
  \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
  \renewcommand*\genacrfullformat}[2]{%
    \glsentrylong{##1}##2\glsacspace{##1}%
    (\protect\firstacronymfont{glsentryshort{##1}})%
  }%
  \renewcommand*\Genacrfullformat}[2]{%
    \Glsentrylong{##1}##2\glsacspace{##1}%
    (\protect\firstacronymfont{glsentryshort{##1}})%
  }%
  \renewcommand*\genplacrfullformat}[2]{%
    \glsentrylongpl{##1}##2\glsacspace{##1}%
    (\protect\firstacronymfont{glsentryshortpl{##1}})%
  }%

```

```

}%
\renewcommand*\Genplacrfullformat}[2]{%
  \Glsentrylongpl{##1}##2\glsacspace{##1}%
  (\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newcommand*\glsacspace}[1]{%
  \settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{##1}})}%
  \ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{
  \ifglsashaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshort{##1}}##2\space
  (\glsentrylong{##1})%
}%
\renewcommand*\Genacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshort{##1}}##2\space
  (\glsentrylong{##1})%
}%
\renewcommand*\genplacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
  (\glsentrylongpl{##1})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
  (\glsentrylongpl{##1})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*\acronymfont}[1]{\textsc{##1}}%

```

```

\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{long-sm-short}%
{%
\GlsUseAcrEntryDispStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{sc-short-long}%
{%
\GlsUseAcrEntryDispStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{sm-short-long}%
{%
\GlsUseAcrEntryDispStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sp-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sp-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sp-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\glsacspace{##1}(\acronymfont{\glsentryshort{##1}})}%
}

```

```

}
\newacronymstyle{long-sc-short-desc}%
{
  \GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{
  \GlsUseAcrStyleDefs{long-sc-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sm-short-desc}%
{
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{short-long-desc}%
{
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{sc-short-long-desc}%
{
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{sm-short-long-desc}%
{
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{

```



```

\GlsUseAcrStyleDefs{sm-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \Glsentrylongpl{\glslabel}\glsinsert
        }%
        \mfirstucMakeUppercase
        {\glsentrylongpl{\glslabel}\glsinsert}%
      }%
    }%
    {%
      \glscapscase
      {%
        \glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \Glsentrylong{\glslabel}\glsinsert
      }%
      \mfirstucMakeUppercase
      {\glsentrylong{\glslabel}\glsinsert}%
    }%
  }%
  }%
  {%
    \glsgenentryfmt
  }%
  {\glscustomtext\glsinsert}%
}%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\glsentrylong{##2}##3\space

```

```

        (\acronymfont{\glsentryshort{##2}})}}%
\renewcommand*{\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{\Glsentrylong{##2}##3\space
        (\acronymfont{\glsentryshort{##2}})}}%
\renewcommand*{\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
        \mfirstucMakeUppercase{\glsentrylong{##2}##3\space
            (\acronymfont{\glsentryshort{##2}})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
    \glslink[##1]{##2}{\glsentrylongpl{##2}##3\space
        (\acronymfont{\glsentryshortpl{##2}})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
    \glslink[##1]{##2}{\Glsentrylongpl{##2}##3\space
        (\acronymfont{\glsentryshortpl{##2}})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
    \glslink[##1]{##2}{%
        \mfirstucMakeUppercase{\glsentrylongpl{##2}##3\space
            (\acronymfont{\glsentryshortpl{##2}})}}}%
\renewcommand*{\glsentryfull}[1]{%
    \glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
    \Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
    \glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
    \Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
    \GlsUseAcrEntryDispStyle{dua}%
}%
{%
    \GlsUseAcrStyleDefs{dua}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentrylong{##1}}}%
    \renewcommand*{\acronymsort}[2]{##2}%
}%
\newacronymstyle{footnote}%
{%
    \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%

```

```

\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\glsHYperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshort{##1}}##2%
  \protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
  \firstacronymfont{\Glsentryshort{##1}}##2%
  \protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
  \mfirstucMakeUppercase{\acronymfont{\glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
  \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}}}%
\renewcommand*{\glsentryfull}[1]{%
  \acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\glsentryfullpl}[1]{%
  \acronymfont{\glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%

```

```

}
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}%
\newacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}%
\newacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*{\GenericAcronymFields}{}%
}

```

```

\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newcommand*{\DefineAcronymSynonyms}{%
  \let\acs\acrshort
  \let\Acs\Acrshort
  \let\acsp\acrshortpl
  \let\Acsp\Acrshortpl
  \let\acl\acrlong
  \let\aclp\acrlongpl
  \let\Acl\Acrlong
  \let\Aclp\Acrlongpl
  \let\acf\acrfull
  \let\acfp\acrfullpl
  \let\Acf\Acrfull
  \let\Acfp\Acrfullpl
  \let\ac\gls
  \let\Ac\Gls
  \let\acp\glspl
  \let\Acp\Glspl
}
\ifglsacrshortcuts
  \DefineAcronymSynonyms
\fi
\newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*{\DefaultNewAcronymDef}{%
  \edef\do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      sort={\the\glsshorttok},%
      text={\the\glsshorttok},%
      first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
      plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      firstplural={\acrfullformat{\noexpand\expandonce\noexpand\@glo@longpl}%
        {\noexpand\expandonce\noexpand\@glo@shortpl}},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      \the\glskeylisttok
    }%
  }%
\let\@org@gls@assign@firstpl\gls@assign@firstpl

```

```

\let\@org@gl@assign@plural\gl@assign@plural
\let\@org@gl@assign@descplural\gl@assign@descplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@gl@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\@org@gl@assign@firstpl
\let\gl@assign@plural\@org@gl@assign@plural
\let\gl@assign@symbolplural\@org@gl@assign@symbolplural
\@gl@noexpand@assign@default@plurals
}
\newcommand*{\@gl@noexpand@assign@default@plurals}{%
  \expandafter\ifx\csname glo@the\gl@labeltok @plural\endcsname\@gl@@default@acr@plural
    \csxdef{glo@the\gl@labeltok @plural}{\expandonce\@glo@shortpl}%
  \fi
  \expandafter\ifx\csname glo@the\gl@labeltok @firstpl\endcsname\@gl@@default@acr@firstplural
    \csxdef{glo@the\gl@labeltok @firstpl}{%
      \expandonce\@glo@longpl\glsspace(\expandonce\@glo@shortpl)}%
    \fi
  \expandafter\ifx\csname glo@the\gl@labeltok @descplural\endcsname\@gl@@default@acr@descplural
    \csxdef{glo@the\gl@labeltok @descplural}{\expandonce\@glo@longpl}%
    \fi
}
\newcommand*{\@gl@@default@acr@plural}{\expandonce\@glo@shortpl}
\newcommand*{\@gl@@default@acr@firstplural}{%
  \expandonce\@glo@longpl\glsspace(\expandonce\@glo@shortpl)%
}
\newcommand*{\@gl@@default@acr@descplural}{\expandonce\@glo@longpl}
\newcommand*{\SetDefaultAcronymStyle}{%
  \@for\@gl@type:=\@gl@sacronymlists\do{%
    \SetDefaultAcronymDisplayStyle{\@gl@type}%
  }%
}
\renewcommand{\newacronym}[4][{}]{%
  \ifx\@gl@sacronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetDefaultAcronymDisplayStyle{\@glo@type}%
  \fi
  \gl@keylisttok{##1}%
  \gl@labeltok{##2}%
  \gl@shorttok{##3}%
  \gl@longtok{##4}%
  \newacronymhook
}

```

```

\DefaultNewAcronymDef
}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
\newcommand*{\acrlinkfootnote}[3]{%
\footnote{\glslink[#1]{#2}{#3}}%
}
\newcommand*{\acrnolinkfootnote}[3]{%
\footnote{#3}%
}
\newcommand*{\SetDescriptionFootnoteAcronymDisplayStyle}[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\ifglsused{\glslabel}%
{%
\acronymfont{\glsgenentryfmt}%
}%
{%
\firstacronymfont{\glsgenentryfmt}%
\ifgls hassymbol{\glslabel}%
{%
\expandafter\protect\expandafter\acrfootnote\expandafter
{\@gls@link@opts}{\@gls@link@label}%
{%
\glsifplural
{\glsentrysymbolplural{\glslabel}}%
{\glsentrysymbol{\glslabel}}%
}%
}%
}%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*{\DescriptionFootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
first={\the\glsshorttok},%
firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
}
}
}

```

```

    longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    symbol={\the\glslongtok},%
    symbolplural={\noexpand\expandonce\noexpand\@glo@longpl},%
    \the\glskeylisttok
  }%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionFootnoteAcronymStyle}{%
  \ifglsacrsmallcaps
    \@gls@deprecated@acr@warning{footnote,smallcaps,description}%
    {footnote-sc-desc}{short-sc-footnote-desc}%
  \else
    \ifglsacrsmaller
      \@gls@deprecated@acr@warning{footnote,smaller,description}%
      {footnote-sm-desc}{short-sm-footnote-desc}%
    \else
      \@gls@deprecated@acr@warning{footnote,description}%
      {footnote-desc}{short-footnote-desc}%
    \fi
  \fi
\renewcommand{\newacronym}[4][]{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetDescriptionFootnoteAcronymDisplayStyle{\@glo@type}%
  \fi
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \glslongtok{##4}%
  \newacronymhook
  \DescriptionFootnoteNewAcronymDef
}%

```



```

\@for\@gls@type:=\@glsacronymlists\do{%
  \SetDescriptionFootnoteAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
  \renewcommand*\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
\else
  \ifglsacrsmaller
    \renewcommand*\acronymfont}[1]{\textsmaller{##1}}%
  \fi
\fi
\ifglsacrdua
  \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
can’t both be set}{}%
\fi
}%
\newcommand*\SetDescriptionDUAAcronymDisplayStyle}[1]{%
  \defglsentryfmt[##1]{\glsgenentryfmt}%
}
\newcommand*\DescriptionDUANewAcronymDef){%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glslongtok},%
      sort={\the\glslongtok},%
      text={\the\glslongtok},%
      first={\the\glslongtok},%
      plural={\noexpand\expandonce\noexpand\@glo@longpl},%
      firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      symbol={\the\glsshorttok},%
      symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      \the\glskeylisttok
    }%
  }%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}

```

```

}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*\SetDescriptionDUAAcronymStyle{%
\ifglsacrsmallcaps
\PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
can’t both be set}{}%
\else
\ifglsacrsmaller
\PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
can’t both be set}{}%
\fi
\fi
\@gls@deprecated@acr@warning{dua,description}{dua-desc}{long-noshort-desc}%
\renewcommand*\newacronym[4][]{%
\ifx\@glsacronymlists\@empty
\def\@glo@type{\acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@glo@type}%
\SetDescriptionDUAAcronymDisplayStyle{\@glo@type}%
\fi
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\DescriptionDUANewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetDescriptionDUAAcronymDisplayStyle{\@gls@type}%
}%
}%
\newcommand*\SetDescriptionAcronymDisplayStyle[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\ifglsused{\glslabel}%
{%
\let\gls@org@insert\glsinsert
\let\glsinsert\@empty
\acronymfont{\glsgenentryfmt}\gls@org@insert
}%
}%
\glsgenentryfmt
\ifglshassymbol{\glslabel}%
{%
\glsifplural

```

```

        {%
        \def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%
        }%
        {%
        \def\@glo@symbol{\glsentrysymbol{\glslabel}}%
        }%
        \space(\protect\firstacronymfont
        {\gls caps case
        {\@glo@symbol}
        {\@glo@symbol}
        {\mfirstucMakeUppercase{\@glo@symbol}}})%
        }%
        {}%
    }%
    }%
    {\gls custom text\gls insert}%
}
}
\newcommand*{\DescriptionNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand
\acrnameformat{\the\glsshorttok}{\the\gls longtok}},%
sort={\the\glsshorttok},%
first={\the\gls longtok},%
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\gls longtok},%
longplural={\the\gls longtok\noexpand\acrpluralsuffix},%
symbol={\noexpand\@glo@text},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
\@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
\@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
\@gls@expand@field{##1}{symbolplural}{##2}%
}%
}

```

```

\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionAcronymStyle}{%
\ifglsacrsmallcaps
\@gls@deprecated@acr@warning{smallcaps,description}%
{long-sc-short-desc}{long-short-sc-desc}%
\else
\ifglsacrsmaller
\@gls@deprecated@acr@warning{smaller,description}%
{long-sm-short-desc}{long-short-sm-desc}%
\else
\@gls@deprecated@acr@warning{description}%
{long-short-desc}{long-short-desc}%
\fi
\fi
\renewcommand{\newacronym}[4][]{%
\ifx\@glsacronymlists\@empty
\def\@glo@type{\acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@glo@type}%
\SetDescriptionAcronymDisplayStyle{\@glo@type}%
\fi
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\DescriptionNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetDescriptionAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
\renewcommand{\acronymfont}[1]{\textsc{##1}}
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
\else
\ifglsacrsmaller
\renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
\fi
\fi
}%
\newcommand*{\SetFootnoteAcronymDisplayStyle}[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\let\gls@org@insert\glsinsert
\let\glsinsert\@empty

```

```

\ifglsused{\glslabel}%
{%
  \acronymfont{\glsgetentryfmt}\gls@org@insert
}%
{%
  \firstacronymfont{\glsgetentryfmt}\gls@org@insert
  \ifglsashaslong{\glslabel}%
  {%
    \expandafter\protect\expandafter\acrfootnote\expandafter
    {\@gls@link@opts}{\@gls@link@label}%
    {%
      \glsifplural
      {\glsentrylongpl{\glslabel}}%
      {\glsentrylong{\glslabel}}%
    }%
  }%
  }%
  {}%
}%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*{\FootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\glsshorttok}},%
    sort={\the\glsshorttok},%
    text={\the\glsshorttok},%
    plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
    first={\the\glsshorttok},%
    firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
    short={\the\glsshorttok},%
    shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
    long={\the\glslongtok},%
    longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    description={\the\glslongtok},%
    descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
    \the\glskeylisttok
  }%
}%
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}

```

```

}%
\def\gls@assign@descplural##1##2{%
  \@gls@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@descplural\@org@gls@assign@descplural
}
\newcommand*\SetFootnoteAcronymStyle{%
  \ifglsacrsmallcaps
    \@gls@deprecated@acr@warning{footnote,smallcaps}{footnote-sc}{short-sc-footnote}%
  \else
    \ifglsacrsmaller
      \@gls@deprecated@acr@warning{footnote,smaller}{footnote-sm}{short-sm-footnote}%
    \else
      \@gls@deprecated@acr@warning{footnote}{footnote}{short-footnote}%
    \fi
  \fi
\renewcommand{\newacronym}[4] []{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetFootnoteAcronymDisplayStyle{\@glo@type}%
  \fi
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \glslongtok{##4}%
  \newacronymhook
  \FootnoteNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
  \SetFootnoteAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
  \renewcommand*\acronymfont[1]{\textsc{##1}}%
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
\else
  \ifglsacrsmaller
    \renewcommand*\acronymfont[1]{\textsmaller{##1}}%
  \fi
\fi
\ifglsacrdua
  \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
  can’t both be set}{}%
\fi
}%
\DeclareRobustCommand*\glsdoparenifnotempty}[2]{%

```

```

\protected@edef\gls@tmp{#1}%
\ifdefempty\gls@tmp
{}%
{%
  \ifx\gls@tmp\@gls@default@value
  \else
  \space (#2{#1})%
  \fi
}%
}
\newcommand*{\SetSmallAcronymDisplayStyle}[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\gls@customtext
{%
\let\gls@org@insert\gls@insert
\let\gls@insert\@empty
\ifglsused{\gls@label}%
{%
\acronymfont{\gls@genentryfmt}\gls@org@insert
}%
{%
\gls@genentryfmt
\ifgls@shassymbol{\gls@label}%
{%
\gls@ifplural
{%
\def\@glo@symbol{\gls@entrysymbolplural{\gls@label}}%
}%
{%
\def\@glo@symbol{\gls@entrysymbol{\gls@label}}%
}%
\space
(\gls@scapscase
{\firstacronymfont{\@glo@symbol}}%
{\firstacronymfont{\@glo@symbol}}%
{\firstacronymfont{\mfirstucMakeUppercase{\@glo@symbol}}})%
}%
}%
}\gls@customtext\gls@insert}%
}%
}
\newcommand*{\SmallNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\gls@labeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\gls@shorttok}},%
sort={\the\gls@shorttok},%

```

```

text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
first={\the\glslongtok},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\noexpand\@glo@first},%
descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
symbol={\the\glsshorttok},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
\@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
\@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
\@gls@expand@field{##1}{descplural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
\@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@descplural\@org@gls@assign@descplural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetSmallAcronymStyle}{%
\ifglsacrsmallcaps
\@gls@deprecated@acr@warning{smallcaps}{long-sc-short}{long-short-sc}%
\else
\@gls@deprecated@acr@warning{smaller}{long-sm-short}{long-short-sm}%
\fi
\renewcommand{\newacronym}[4][]{%
\ifx\@glsacronymlists\@empty
\def\@glo@type{acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@glo@type}%
\SetSmallAcronymDisplayStyle{\@glo@type}%
\fi
}

```



```

\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\SmallNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
  \SetSmallAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
  \renewcommand*\acronymfont}[1]{\textsc{##1}}
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
\else
  \renewcommand*\acronymfont}[1]{\textsmaller{##1}}
\fi
\ifglsacrdua
  \ifglsacrsmallcaps
    \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
      can’t both be set}{}%
  \else
    \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
      can’t both be set}{}%
  \fi
\fi
}%
\newcommand*\SetDUADisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*\DUANewAcronymDef{%
  \edef\@do@newglossaryentry{%
    \noexpandnewglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      text={\the\glslongtok},%
      first={\the\glslongtok},%
      plural={\noexpand\expandonce\noexpand\@glo@longpl},%
      firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      symbol={\the\glsshorttok},%
      symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      \the\glskeylisttok
    }%
  }%
}

```

```

\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@gls@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
\let\gls@assign@descplural\@org@gls@assign@descplural
}
\newcommand*{\SetDUASStyle}{%
  \@gls@deprecated@acr@warning{dua}{dua}{long-noshort}%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@glsacronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetDUADisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \DUANewAcronymDef
  }%
  \@for\@gls@type:=\@glsacronymlists\do{%
    \SetDUADisplayStyle{\@gls@type}%
  }%
}
\newcommand*{\@gls@deprecated@acr@warning}[3]{%
  \GlossariesWarning{Acronym style option #1 deprecated.
  Use \string\setacronymstyle{#2} with just glossaries.sty or
  use \string\setabbreviationstyle[acronym]{#3} with
  glossaries-extra.sty}%
}
\newcommand*{\SetAcronymStyle}{%
  \SetDefaultAcronymStyle

```

```

\ifglsacrdescription
  \ifglsacrfootnote
    \SetDescriptionFootnoteAcronymStyle
  \else
    \ifglsacrdua
      \SetDescriptionDUAAcronymStyle
    \else
      \SetDescriptionAcronymStyle
    \fi
  \fi
\else
  \ifglsacrfootnote
    \SetFootnoteAcronymStyle
  \else
    \ifthenelse{\boolean{glsacrsmallcaps}\OR
      \boolean{glsacrsmaller}}{%
      \SetSmallAcronymStyle
    }%
    \ifglsacrdua
      \SetDUASyle
    \fi
  }%
\fi
\fi
}
\SetAcronymStyle
\newcommand*{\SetCustomDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*{\CustomAcronymFields}{%
  name={\the\glsshorttok},%
  description={\the\glslongtok},%
  first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
  firstplural={\acrfullformat
    {\noexpand\glsentrylongpl{\the\glslabeltok}}%
    {\noexpand\glsentryshortpl{\the\glslabeltok}}},%
  text={\the\glsshorttok},%
  plural={\the\glsshorttok\noexpand\acrpluralsuffix}%
}
\newcommand*{\CustomNewAcronymDef}{%
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    }
  }
}

```

```

        user1={\the\glsshorttok},%
        user2={\the\glsshorttok\noexpand\acrpluralsuffix},%
        user3={\the\glslongtok},%
        user4={\the\glslongtok\noexpand\acrpluralsuffix},%
        \CustomAcronymFields,%
        \the\glskeylisttok
    }%
}%
\do@newglossaryentry
}
\newcommand*{\SetCustomStyle}{%
\renewcommand{\newacronym}[4][{}]{%
\ifx\@glsacronymlists\@empty
\def\@gls@type{\acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@gls@type}%
\SetCustomDisplayStyle{\@gls@type}%
\fi
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\CustomNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetCustomDisplayStyle{\@gls@type}%
}%
}
\RequirePackage{glossary-hypernav}[=v4.49]
\@gls@loadlist
\@gls@loadlong
\@gls@loadsuper
\@gls@loadtree
\ifx\@glossary@default@style\relax
\else
\setglossarystyle{\@glossary@default@style}
\fi
\newcommand*{\showgloparent}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
}
\newcommand*{\showglolevel}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
}
\newcommand*{\showglotext}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
}
\newcommand*{\showgloplural}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@plural\endcsname
}
}

```

```

\newcommand*\showglofirst}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@first\endcsname
}
\newcommand*\showglofirstpl}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@firstpl\endcsname
}
\newcommand*\showgloftype}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@ftype\endcsname
}
\newcommand*\showglocounter}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@counter\endcsname
}
\newcommand*\showglouser}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@user\endcsname
}
\newcommand*\showglouserii}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@userii\endcsname
}
\newcommand*\showglouseriii}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@useriii\endcsname
}
\newcommand*\showglouseriv}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@useriv\endcsname
}
\newcommand*\showglouserv}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@userv\endcsname
}
\newcommand*\showglouservi}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@uservi\endcsname
}
\newcommand*\showglongame}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@name\endcsname
}
\newcommand*\showglodesc}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@desc\endcsname
}
\newcommand*\showglodescplural}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@descplural\endcsname
}
\newcommand*\showglosort}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@sort\endcsname
}
\newcommand*\showglosymbol}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@symbol\endcsname
}
\newcommand*\showglosymbolplural}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@symbolplural\endcsname
}
\newcommand*\showgloshort}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@short\endcsname
}

```

```

}
\newcommand*\showglolong}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname
}
\newcommand*\showgloindex}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname
}
\newcommand*\showgloflag}[1]{%
  \expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname
}
\newcommand*\showglooclist}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@oclist\endcsname
}
\newcommand*\showglofield}[2]{%
  \csshow{glo@\glsdetoklabel{#1}@#2}%
}
\newcommand*\showacronymlists}{%
  \show\@glsacronymlists
}
\newcommand*\showglossaries}{%
  \show\@glo@types
}
\newcommand*\showglossaryin}[1]{%
  \expandafter\show\csname @glo@type@#1@in\endcsname
}
\newcommand*\showglossaryout}[1]{%
  \expandafter\show\csname @glo@type@#1@out\endcsname
}
\newcommand*\showglossarytitle}[1]{%
  \expandafter\show\csname @glo@type@#1@title\endcsname
}
\newcommand*\showglossarycounter}[1]{%
  \expandafter\show\csname @glo@type@#1@counter\endcsname
}
\newcommand*\showglossaryentries}[1]{%
  \expandafter\show\csname glist@#1\endcsname
}
\csname ifglscompatible-2.07\endcsname
  \RequirePackage{glossaries-compatible-207}[v4.49]
\fi

```

8.19 Rollback v4.49 (glossaries-accsupp-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-accsupp}[2021/11/01 v4.49 (NLCT)
  Experimental glossaries accessibility]
\DeclareOption*\PassOptionsToPackage{\CurrentOption}{glossaries}}
\ProcessOptions
\@ifpackageloaded{glossaries-extra}

```

```

{%
\ifx\@glsxtr@doaccsupp\empty
\GlossariesWarning{The 'glossaries-accsupp'
package has been loaded\MessageBreak
after the 'glossaries-extra' package. This\MessageBreak
can cause a failure to integrate both packages. \MessageBreak
Either use the 'accsupp' option when you load\MessageBreak
'glossaries-extra' or load 'glossaries-accsupp'\MessageBreak
before loading 'glossaries-extra'}%
\fi
}
{}
\def\compatibleglossentry#1#2{%
\toks@{#2}%
\protected@edef\@do@glossentry{%
\noexpand\accsuppglossaryentryfield{#1}%
{ \noexpand\glsnamefont
{\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@name\endcsname}}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@desc\endcsname}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@symbol\endcsname}%
{\the\toks@}%
}%
\@do@glossentry
}
\def\compatiblesubglossentry#1#2#3{%
\toks@{#3}%
\protected@edef\@do@subglossentry{%
\noexpand\accsuppglossarysubentryfield{\number#1}%
{#2}%
{ \noexpand\glsnamefont
{\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@name\endcsname}}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@desc\endcsname}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@symbol\endcsname}%
{\the\toks@}%
}%
\@do@subglossentry
}
\RequirePackage{glossaries}[v4.49]
\providecommand{\gls@accsupp@engine}{accsupp}
\providecommand{\gls@accessibility}[4]{#4}
\ifdefstring\gls@accsupp@engine{accsupp}
{
\RequirePackage{accsupp}
\renewcommand{\gls@accessibility}[4]{%
\BeginAccSupp{#1,#2={#3}}#4\EndAccSupp{}}
}
}
{}
\newcommand{\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}{useriiaccess},%
  {user3access}{useriiiaccess},%
  {user4access}{userivaccess},%
  {user5access}{uservaccess},%
  {user6access}{userviaccess}%
}
\def\@gls@noaccess{\relax}
\let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
\renewcommand*{\@newglossaryentryprehook}{%
  \@gls@oldnewglossaryentryprehook
  \def\@glo@access{\relax}%
  \def\@glo@textaccess{\@glo@access}%
  \def\@glo@firstaccess{\@glo@access}%
  \def\@glo@pluralaccess{\@glo@textaccess}%
  \def\@glo@firstpluralaccess{\@glo@pluralaccess}%
  \def\@glo@symbolaccess{\relax}%
  \def\@glo@symbolpluralaccess{\@glo@symbolaccess}%
  \def\@glo@descaccess{\relax}%
  \def\@glo@descpluralaccess{\@glo@descaccess}%
  \def\@glo@shortaccess{\relax}%
  \def\@glo@shortpluralaccess{\@glo@shortaccess}%
  \def\@glo@longaccess{\relax}%
  \def\@glo@longpluralaccess{\@glo@longaccess}%
  \def\@glo@useriaccess{\relax}%
  \def\@glo@useriiaccess{\relax}%
  \def\@glo@useriiiaccess{\relax}%
}

```

```

\def@glo@userivaccess{\relax}%
\def@glo@uservaccess{\relax}%
\def@glo@userviaccess{\relax}%
}
\let@gls@oldnewglossaryentryposthook@newglossaryentryposthook
\renewcommand*{@newglossaryentryposthook}{%
  @gls@oldnewglossaryentryposthook
  \expandafter
  \protected@xdef\csname glo@%glo@label @access\endcsname{%
    @glo@access}%
  \expandafter
  \protected@xdef\csname glo@%glo@label @textaccess\endcsname{%
    @glo@textaccess}%
  \expandafter
  \protected@xdef\csname glo@%glo@label @firstaccess\endcsname{%
    @glo@firstaccess}%
  \expandafter
  \protected@xdef\csname glo@%glo@label @pluralaccess\endcsname{%
    @glo@pluralaccess}%
  \expandafter
  \protected@xdef\csname glo@%glo@label @firstpluralaccess\endcsname{%
    @glo@firstpluralaccess}%
  \expandafter
  \protected@xdef\csname glo@%glo@label @symbolaccess\endcsname{%
    @glo@symbolaccess}%
  \expandafter
  \protected@xdef\csname glo@%glo@label @symbolpluralaccess\endcsname{%
    @glo@symbolpluralaccess}%
  \expandafter
  \protected@xdef\csname glo@%glo@label @descaccess\endcsname{%
    @glo@descaccess}%
  \expandafter
  \protected@xdef\csname glo@%glo@label @descpluralaccess\endcsname{%
    @glo@descpluralaccess}%
  \expandafter
  \protected@xdef\csname glo@%glo@label @shortaccess\endcsname{%
    @glo@shortaccess}%
  \expandafter
  \protected@xdef\csname glo@%glo@label @shortpluralaccess\endcsname{%
    @glo@shortpluralaccess}%
  \expandafter
  \protected@xdef\csname glo@%glo@label @longaccess\endcsname{%
    @glo@longaccess}%
  \expandafter
  \protected@xdef\csname glo@%glo@label @longpluralaccess\endcsname{%
    @glo@longpluralaccess}%
  \expandafter
  \protected@xdef\csname glo@%glo@label @useriaccess\endcsname{%
    @glo@useriaccess}%
  \expandafter

```

```

\protected@xdef\csname glo@\glo@label @useriaccess\endcsname{%
  \@glo@useriaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @useriiiaccess\endcsname{%
  \@glo@useriiiaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @userivaccess\endcsname{%
  \@glo@userivaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @uservaccess\endcsname{%
  \@glo@uservaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @userviaccess\endcsname{%
  \@glo@userviaccess}%
}
\newcommand*{\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*\glseentrylongpluralaccess}[1]{%
  \@gls@entry@field{#1}{longpluralaccess}%
}
\newcommand*\glseentryuseriaccess}[1]{%
  \@gls@entry@field{#1}{useriaccess}%
}
\newcommand*\glseentryuseriiaccess}[1]{%
  \@gls@entry@field{#1}{useriiaccess}%
}
\newcommand*\glseentryuseriiiaccess}[1]{%
  \@gls@entry@field{#1}{useriiiaccess}%
}
\newcommand*\glseentryuserivaccess}[1]{%
  \@gls@entry@field{#1}{userivaccess}%
}
\newcommand*\glseentryuservaccess}[1]{%
  \@gls@entry@field{#1}{uservaccess}%
}
\newcommand*\glseentryuserviaccess}[1]{%
  \@gls@entry@field{#1}{userviaccess}%
}
\newcommand*\glsfieldaccsupp}[4]{%
  \ifdef\glscategory
  {%
    \ifcsdef{glsxtr\glscategory{#4}#3accsupp}%
    {\csname glsxtr\glscategory{#4}#3accsupp\endcsname{#1}{#2}}%
    {%
      \ifcsdef{glsxtr\glscategory{#4}accsupp}%
      {\csname glsxtr\glscategory{#4}accsupp\endcsname{#1}{#2}}%
      {%
        \ifcsdef{gls#3accsupp}%
        {\csname gls#3accsupp\endcsname{#1}{#2}}%
        {\glsaccsupp{#1}{#2}}%
      }%
    }%
  }%
  {%
    \ifcsdef{gls#3accsupp}%
    {\csname gls#3accsupp\endcsname{#1}{#2}}%
    {\glsaccsupp{#1}{#2}}%
  }%
}
\newcommand*\xglsfieldaccsupp}[1]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsfieldaccsupp\expandafter{\@gls@replacementtext}%
}
\newcommand*\glsshortaccsupp}[2]{\glsaccessibility{E}{#1}{#2}}
\newcommand*\glsshortplaccsupp}{\glsshortaccsupp}
\newcommand*\glsaccsupp}[2]{\glsaccessibility{ActualText}{#1}{#2}}

```

```

\newcommand*\xglsaccsupp}[2]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsaccsupp\expandafter{\@gls@replacementtext}{#2}%
}
\newcommand*\@gls@access@display}[2]{%
  \protected@edef\@glo@access{#2}%
  \ifx\@glo@access\@gls@noaccess
    #1%
  \else
    \xglsaccsupp{\@glo@access}{#1}%
  \fi
}
\newcommand*\@gls@fieldaccess@display}[4]{%
  \protected@edef\@glo@access{#4}%
  \ifdefequal\@glo@access\@gls@noaccess
    {#3}%
  {\expandafter\glsfieldaccsupp\expandafter{\@glo@access}{#3}{#2}{#1}}%
}
\newrobustcmd*\glsnameaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@access}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{name}{#1}{\glsentryaccess{#2}}}%
}
\newrobustcmd*\gls@textaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@textaccess}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{text}{#1}{\glsentrytextaccess{#2}}}%
}
\newrobustcmd*\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*\gls@symbolaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@symbolaccess}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{symbol}{#1}{\glsentrysymbolaccess{#2}}}%
}
\newrobustcmd*\gls@symbolpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@symbolpluralaccess}%

```

```

    {#1}%
    {\@gls@fieldaccess@display{#2}{symbolplural}{#1}{\glsentrysymbolpluralaccess{#2}}}%
  }
\newrobustcmd*{\glsdescriptionaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@descaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{desc}{#1}{\glsentrydescaccess{#2}}}%
}
\newrobustcmd*{\glsdescriptionpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@descpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{descplural}{#1}{\glsentrydescpluralaccess{#2}}}%
}
\newrobustcmd*{\glsshortaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@shortaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{short}{#1}{\glsentryshortaccess{#2}}}%
}
\newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@shortpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glsentryshortpluralaccess{#2}}}%
}
\newrobustcmd*{\glslongaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@longaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{long}{#1}{\glsentrylongaccess{#2}}}%
}
\newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@longpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{longpl}{#1}{\glsentrylongpluralaccess{#2}}}%
}
\newrobustcmd*{\glsuseriaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@useriaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useri}{#1}{\glsentryuseriaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@useriiaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{userii}{#1}{\glsentryuseriiaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiiaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@useriiiaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useriii}{#1}{\glsentryuseriiiaccess{#2}}}%
}
\newrobustcmd*{\glsuserivaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@userivaccess}%

```

```

    {#1}%
    {\@gls@fieldaccess@display{#2}{useriv}{#1}{\glsentryuserivaccess{#2}}}%
  }
\newrobustcmd*{\glsuservaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@uservaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{userv}{#1}{\glsentryuservaccess{#2}}}%
}
\newrobustcmd*{\glsuserviaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@userviaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{uservi}{#1}{\glsentryuserviaccess{#2}}}%
}
\newrobustcmd*{\glsaccessdisplay}[3]{%
  \ifcsundef{gls#1accessdisplay}%
  {%
    \PackageError{glossaries-accsupp}{No accessibility support
      for key ‘#1’}{%
    }%
    {%
      \csname gls#1accessdisplay\endcsname{#2}{#3}%
    }%
  }
}
\renewcommand*{\@gls@default@entryfmt}[2]{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          #2{\glspluralaccessdisplay
            {\glsentryplural{\glslabel}}{\glslabel}}%
          {\glsdescriptionpluralaccessdisplay
            {\glsentrydescplural{\glslabel}}{\glslabel}}%
          {\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
    {%

```



```

}%
{%
  \MakeUppercase{%
    #1{\glsfirstaccessdisplay
      {\glsentryfirst{\glslabel}}{\glslabel}}%
      {\glsdescriptionaccessdisplay
        {\glsentrydesc{\glslabel}}{\glslabel}}%
      {\glsymbolaccessdisplay
        {\glsentrysymbol{\glslabel}}{\glslabel}}%
      {\glsinsert}}%
  }%
}%
}%
}%
{%
  \ifglsused{\glslabel}%
  {%
    #2{\glscustomtext}%
    {\glsdescriptionaccessdisplay
      {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
      {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
  {%
    #1{\glscustomtext}%
    {\glsdescriptionaccessdisplay
      {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
      {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
}%
}
\renewcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          \glspluralaccessdisplay
            {\glsentryplural{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glsfirstpluralaccessdisplay
            {\glsentryfirstplural{\glslabel}}{\glslabel}%

```

```

    \glsinsert
  }%
}%
{%
  \ifglsused\glslabel
  {%
    \glspluralaccessdisplay
      {\Glsentryplural{\glslabel}}{\glslabel}%
    \glsinsert
  }%
  {%
    \glsfirstpluralaccessdisplay
      {\Glsentryfirstplural{\glslabel}}{\glslabel}%
    \glsinsert
  }%
}%
{%
  \ifglsused\glslabel
  {%
    \glspluralaccessdisplay
      {\mfirstucMakeUppercase{\glsentryplural{\glslabel}}}%
      {\glslabel}%
    \mfirstucMakeUppercase{\glsinsert}%
  }%
  {%
    \glsfirstpluralaccessdisplay
      {\mfirstucMakeUppercase{\glsentryfirstplural{\glslabel}}}%
      {\glslabel}%
    \mfirstucMakeUppercase{\glsinsert}%
  }%
}%
}%
{%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      \glstextaccessdisplay{\glsentrytext{\glslabel}}{\glslabel}%
      \glsinsert
    }%
    {%
      \glsfirstaccessdisplay{\glsentryfirst{\glslabel}}{\glslabel}%
      \glsinsert
    }%
  }%
}%
{%
  \ifglsused\glslabel
  {%
    \glstextaccessdisplay{\Glsentrytext{\glslabel}}{\glslabel}%
    \glsinsert
  }%
}

```

```

    }%
    {%
      \glsfirstaccessdisplay{\Glsentryfirst{\glslabel}}{\glslabel}%
      \glsinsert
    }%
  }%
  {%
    \ifglsused\glslabel
    {%
      \glstextaccessdisplay
      {\mfirstucMakeUppercase{\glsentrytext{\glslabel}}}{\glslabel}%
      \mfirstucMakeUppercase{\glsinsert}%
    }%
    {%
      \glsfirstaccessdisplay
      {\mfirstucMakeUppercase{\glsentryfirst{\glslabel}}}{\glslabel}%
      \mfirstucMakeUppercase{\glsinsert}%
    }%
  }%
}
}
}
}
\renewcommand*{\glsgenacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglsused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \acronymfont
          {\glsshortpluralaccessdisplay
            {\glsentryshortpl{\glslabel}}{\glslabel}}%
          \glsinsert
        }%
        {%
          \acronymfont
          {\glsshortpluralaccessdisplay
            {\Glsentryshortpl{\glslabel}}{\glslabel}}%
          \glsinsert
        }%
        {%
          \mfirstucMakeUppercase
          {\acronymfont
            {\glsshortpluralaccessdisplay
              {\glsentryshortpl{\glslabel}}{\glslabel}}%
          }%

```

```

        \glsinsert}%
    }%
}%
{%
    \glscapscase
    {%
        \acronymfont
        {\glsshortaccessdisplay{\glsentryshort{\glslabel}}{\glslabel}}%
        \glsinsert
    }%
    {%
        \acronymfont
        {\glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
        \glsinsert
    }%
    {%
        \mfirstucMakeUppercase
        {\acronymfont{%
            \glsshortaccessdisplay{\glsentryshort{\glslabel}}{\glslabel}}%
            \glsinsert}%
    }%
}%
}%
{%
    \glsifplural
    {%
        \glscapscase
        {%
            \genplacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \Genplacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \mfirstucMakeUppercase
            {\genplacrfullformat{\glslabel}{\glsinsert}}%
        }%
    }%
}%
{%
    \glscapscase
    {%
        \genacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
        \Genacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
        \mfirstucMakeUppercase
        {\genacrfullformat{\glslabel}{\glsinsert}}%
    }%
}%

```

```

    }%
  }%
}%
{%
  \glscustomtext
}%
}
\renewcommand*{\genacrfullformat}[2]{%
  \glslongaccessdisplay{\glsentrylong{#1}}{#1}#2\space
  (\glsshortaccessdisplay{\protect\firstacronymfont{\glsentryshort{#1}}}{#1})%
}
\renewcommand*{\Genacrfullformat}[2]{%
  \glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space
  (\glsshortaccessdisplay{\protect\firstacronymfont{\Glsentryshort{#1}}}{#1})%
}
\renewcommand*{\genplacrfullformat}[2]{%
  \glslongpluralaccessdisplay{\glsentrylongpl{#1}}{#1}#2\space
  (\glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
}
\renewcommand*{\Genplacrfullformat}[2]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}#2\space
  (\glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\Glsentryshortpl{#1}}}{#1})%
}
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\glsentryshort{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@Acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glsapsaps\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
}

```

```

}%
\glspostlinkhook
}
\def\@ACRshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glsshortaccessdisplay
{\MakeUppercase{\glstryshort{#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{\glstrylong{#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

```



```

\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsfirstpluralaccessdisplay{\mfirstucMakeUppercase{\glsentryfirstplural{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glsentryname{#2}}{#2}#3}%
}
\def\@Glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\Glsentryname{#2}}{#2}#3}%
}
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsnameaccessdisplay{\mfirstucMakeUppercase{\glsentryname{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\glsentrydesc{#2}}{#2}#3}%
}
\def\@Glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
}
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsdescriptionaccessdisplay{\mfirstucMakeUppercase{\glsentrydesc{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\glsentrydescplural{#2}}{#2}#3}%
}
\def\@Glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
}
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsdescriptionpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrydescplural{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
}
\def\@Glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}
\def\@GLSsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsymbolaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbol{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsymbolplural@#1#2[#3]{%

```

```

    \@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\glstentrysymbolplural{#2}}{#2}#3}%
  }
\def\@Glsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\glstentrysymbolplural{#2}}{#2}#3}%
}
\def\@GLSsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glssymbolpluralaccessdisplay{\mfirstucMakeUppercase{\glstentrysymbolplural{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glstentryuseri{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriaccessdisplay{\mfirstucMakeUppercase{\glstentryuseri{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glstentryuserii{#2}}{#2}#3}%
}
\def\@Glsuser@ii#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glsentryuserii{#2}}{#2}#3}%
}
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiaccessdisplay{\mfirstucMakeUppercase{\glstentryuserii{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\glstentryuseriii{#2}}{#2}#3}%
}
\def\@Glsuser@iii#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glsentryuseriii{#2}}{#2}#3}%
}
\def\@GLSuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiiaccessdisplay{\mfirstucMakeUppercase{\glstentryuseriii{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glstentryuseriv{#2}}{#2}#3}%
}
\def\@Glsuser@iv#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
}
\def\@GLSuseriv@#1#2[#3]{%

```

```

\@gls@field@link{#1}{#2}%
  {\glsuserivaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriv{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@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}%
{
  \ifglschaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat}[2]{%
  \glsshortaccessdisplay
  {\protect\firstacronymfont{\glentryshort{##1}}}{##1}##2\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*\Genacrfullformat}[2]{%
  \glsshortaccessdisplay
  {\protect\firstacronymfont{\Glsentryshort{##1}}}{##1}##2\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*\genplacrfullformat}[2]{%
  \glsshortpluralaccessdisplay
  {\protect\firstacronymfont{\glentryshortpl{##1}}}{##1}##2\space
  (\glslongpluralaccessdisplay
  {\glentrylongpl{##1}}{##1})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
  \glsshortpluralaccessdisplay
  {\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2\space
  (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})%
}%
\renewcommand*\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\renewacronymstyle{long-short-desc}%
{
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{
  \GlsUseAcrStyleDefs{long-short}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%

```

```

        \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
    }
\renewacronymstyle{long-sc-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sc-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
  }
\renewacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
  }
\renewacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
  }
\renewacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
  }

```

```

}
\renewacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glslongpluralaccessdisplay{\glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay{\Glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay
            {\mfirstucMakeUppercase{\glsentrylongpl{\glslabel}}}{\glslabel}%
            {\mfirstucMakeUppercase{\glsinsert}}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \glslongaccessdisplay{\glsentrylong{\glslabel}}{\glslabel}\glsinsert
      }%
      {%
        \glslongaccessdisplay{\Glsentrylong{\glslabel}}{\glslabel}\glsinsert
      }%
      {%
        \glslongaccessdisplay
          {\mfirstucMakeUppercase
            {\glsentrylong{\glslabel}\glsinsert}}{\glslabel}%
          {\mfirstucMakeUppercase{\glsinsert}}%
      }%
    }%
  }%
}

```

```

    }%
  }%
  {%
    \glsgenentryfmt
  }%
}%
{\glscustomtext\glsinsert}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay
    {\mfirstucMakeUppercase{\glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\Glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\mfirstucMakeUppercase{\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\glsentryfull}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\glsentryfullpl}[1]{%

```



```

\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}\space
(\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*\Glsentryfullpl[1]{%
\glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space
(\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*\acronymentry[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}%
\renewcommand*\acronymsort[2]{##1}%
\renewcommand*\acronymfont[1]{##1}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\renewacronymstyle{dua-desc}%
{%
\GlsUseAcrEntryDispStyle{dua}%
}%
{%
\GlsUseAcrStyleDefs{dua}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymentry[1]{%
\glslongaccessdisplay{\acronymfont{\glsentrylong{##1}}}{##1}%
\renewcommand*\acronymsort[2]{##2}%
}%
\renewacronymstyle{footnote}%
{%
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*\genacrfullformat[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}{##1}}%
}%
\renewcommand*\Genacrfullformat[2]{%
\glsshortaccessdisplay
{\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}{##1}}%
}%
\renewcommand*\genplacrfullformat[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}}%
}%
\renewcommand*\Genplacrfullformat[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}}%
}

```

```

}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glspuralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##2}}}{##2}##3\space
    (\glslongaccessdisplay{\glentrylong{##2}}{##2})}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##2}}}{##2}##3\space
    (\glslongaccessdisplay{\glentrylong{##2}}{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortaccessdisplay
      {\mfirstucMakeUppercase
        {\acronymfont{\glentryshort{##2}}}{##2}##3\space
        (\glslongaccessdisplay{\glentrylong{##2}}{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\glentryshortpl{##2}}}{##2}##3\space
      (\glslongpluralaccessdisplay{\glentrylongpl{##2}}{##2})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\glentryshortpl{##2}}}{##2}##3\space
      (\glslongpluralaccessdisplay{\glentrylongpl{##2}}{##2})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\mfirstucMakeUppercase
        {\acronymfont{\glentryshortpl{##2}}}{##2}##3\space
        (\glslongpluralaccessdisplay{\glentrylongpl{##2}}{##2})}}}%
\renewcommand*{\glentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1})}%
\renewcommand*{\Glentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1})}%
\renewcommand*{\glentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\glentryshortpl{##1}}}{##1}\space
    (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})}%
\renewcommand*{\Glentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\glentryshortpl{##1}}}{##1}\space
    (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})}%

```

```

}
\renewacronymstyle{footnote-sc}%
{
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glstextup{\glspluralsuffix}}%
}%
\renewacronymstyle{footnote-sm}%
{
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}%
\renewacronymstyle{footnote-desc}%
{
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sc-desc}%
{
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sm-desc}%
{
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}

```

```

}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
  }
  \newcommand*\glsdefaultshortaccess}[2]{#1}
  \renewcommand*\newacronymhook{%
    \protected@edef\@gls@keylist{%
      shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
      shortpluralaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
      \the\glskeylisttok}%
    \expandafter\glskeylisttok\expandafter{\@gls@keylist}%
  }
  \renewcommand*\DefaultNewAcronymDef{%
    \edef\@do@newglossaryentry{%
      \noexpand\newglossaryentry{\the\glslabeltok}%
      {%
        type=\acronymtype,%
        name={\the\glsshorttok},%
        description={\the\glslongtok},%
        descriptionaccess=\relax,%
        text={\the\glsshorttok},%
        access={\noexpand\@glo@textaccess},%
        sort={\the\glsshorttok},%
        short={\the\glsshorttok},%
        shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
        shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
        long={\the\glslongtok},%
        longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
        descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
        first={\noexpand\glslongaccessdisplay
          {\the\glslongtok}{\the\glslabeltok}\space
          (\noexpand\glsshortaccessdisplay
            {\the\glsshorttok}{\the\glslabeltok})},%
        plural={\the\glsshorttok\acrpluralsuffix},%
        firstplural={\noexpand\glslongpluralaccessdisplay
          {\noexpand\@glo@longpl}{\the\glslabeltok}\space
          (\noexpand\glsshortpluralaccessdisplay
            {\noexpand\@glo@shortpl}{\the\glslabeltok})},%
        firstaccess=\relax,%
        firstpluralaccess=\relax,%
        textaccess={\noexpand\@glo@shortaccess},%
        \the\glskeylisttok
      }%
    }%
  }
  \let\@org@gls@assign@firstpl\gls@assign@firstpl

```

```

\let\org@gl@assign@plural\gl@assign@plural
\let\org@gl@assign@descplural\gl@assign@descplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@gl@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\org@gl@assign@firstpl
\let\gl@assign@plural\org@gl@assign@plural
\let\gl@assign@symbolplural\org@gl@assign@symbolplural
\@gl@noexpand@assign@default@plurals
}
\renewcommand*{\DescriptionFootnoteNewAcronymDef}{%
\edef\do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\gl@labeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\gl@shorttok}},%
    sort={\the\gl@shorttok},%
    text={\the\gl@shorttok},%
    short={\the\gl@shorttok},%
    shortplural={\the\gl@shorttok\noexpand\acrpluralsuffix},%
    shortaccess={\gl@defaultshortaccess{\the\gl@longtok}{\the\gl@shorttok}},%
    long={\the\gl@longtok},%
    longplural={\the\gl@longtok\noexpand\acrpluralsuffix},%
    access={\noexpand\@glo@textaccess},%
    plural={\the\gl@shorttok\noexpand\acrpluralsuffix},%
    symbol={\the\gl@longtok},%
    symbolplural={\the\gl@longtok\noexpand\acrpluralsuffix},%
    firstpluralaccess=\relax,
    textaccess={\noexpand\@glo@shortaccess},%
    \the\gl@keylisttok
  }%
}%
\let\org@gl@assign@firstpl\gl@assign@firstpl
\let\org@gl@assign@plural\gl@assign@plural
\let\org@gl@assign@symbolplural\gl@assign@symbolplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@symbolplural##1##2{%
  \@gl@expand@field{##1}{symbolplural}{##2}%
}

```

```

}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\DescriptionNewAcronymDef}{%
\edef\do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand
\acronymformat{\the\glsshorttok}{\the\glslongtok}},%
access={\noexpand\@glo@textaccess},%
sort={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
first={\the\glslongtok},%
firstaccess=\relax,
firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
text={\the\glsshorttok},%
textaccess={\the\glslongtok},%
plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
symbol={\noexpand\@glo@text},%
symbolaccess={\noexpand\@glo@textaccess},%
symbolplural={\noexpand\@glo@plural},%
firstpluralaccess=\relax,
textaccess={\noexpand\@glo@shortaccess},%
\the\glskeylisttok}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
\@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
\@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
\@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}

```

```

\renewcommand*{\FootnoteNewAcronymDef}{%
\edef\do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
text={\the\glsshorttok},%
textaccess={\the\glslongtok},%
access={\noexpand\@glo@textaccess},%
plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\the\glslongtok},%
descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
\@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
\@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
\@gls@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@descplural\@org@gls@assign@descplural
}
\renewcommand*{\SmallNewAcronymDef}{%
\edef\do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
access={\noexpand\@glo@symbolaccess},%
sort={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%

```

```

text={\noexpand\@glo@short},%
textaccess={\noexpand\@glo@shortaccess},%
plural={\noexpand\@glo@shortpl},%
first={\the\glslongtok},%
firstaccess=\relax,
firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\noexpand\@glo@first},%
descriptionplural={\noexpand\@glo@firstplural},%
symbol={\the\glsshorttok},%
symbolaccess={\the\glslongtok},%
symbolplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@@gls@expand@field{##1}{descplural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@descplural\@org@gls@assign@descplural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\glsshortaccesskey}{\glsshortkey access}%
\newcommand*{\glsshortpluralaccesskey}{\glsshortpluralkey access}%
\newcommand*{\glslongaccesskey}{\glslongkey access}%
\newcommand*{\glslongpluralaccesskey}{\glslongpluralkey access}%
\newcommand*{\showglongnameaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@access\endcsname
}
\newcommand*{\showglotextaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@textaccess\endcsname
}
\newcommand*{\showglopluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@pluralaccess\endcsname
}
\newcommand*{\showglofirstaccess}[1]{%

```



```

\expandafter\show\csname glo@glstetoklabel{#1}@firstaccess\endcsname
}
\newcommand*\showglofirstpluralaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@firstpluralaccess\endcsname
}
\newcommand*\showglosymbolaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@symbolaccess\endcsname
}
\newcommand*\showglosymbolpluralaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@symbolpluralaccess\endcsname
}
\newcommand*\showglodescaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@descaccess\endcsname
}
\newcommand*\showglodescpluralaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@descpluralaccess\endcsname
}
\newcommand*\showgloshortaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@shortaccess\endcsname
}
\newcommand*\showgloshortpluralaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@shortpluralaccess\endcsname
}
\newcommand*\showglolongaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@longaccess\endcsname
}
\newcommand*\showglolongpluralaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@longpluralaccess\endcsname
}
}

```

8.20 Rollback v4.49 (glossaries-babel-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-babel}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
\ForEachTrackedDialect{\this@dialect}{%
\IfTrackedLanguageFileExists{\this@dialect}%
{glossaries-}% prefix
{.ldf}%
}%
\RequireGlossariesLang{\CurrentTrackedTag}%
}%
}%
\PackageWarningNoLine{glossaries}%
{No language module detected for ‘\this@dialect’.\MessageBreak

```

```

        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~n\string\setentrycounter
      {\noexpand\glscounter}%
      \expandafter\string\csname#1\endcsname
      \expandafter@gobble\string\{\string" ^^J
    :close \string"\expandafter@gobble\string\}\string" ^^J
    :attr \string"#1\string")}}
  \fi
\renewcommand*\GlsAddXdyCounters[1]{%
  \GlossariesWarning{\string\GlsAddXdyCounters\space not available
    in compatibility mode.}%
}
\GlsAddXdyAttribute{glsnumberformat}
\GlsAddXdyAttribute{textrm}
\GlsAddXdyAttribute{textsf}
\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glsnumber}
\GlsAddXdyAttribute{hyperrm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hypertt}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}

```

```

\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}
\ifglxindy
  \renewcommand*{\GlsAddXdyLocation}[2]{%
    \edef\@xdyuserlocationdefs{%
      \@xdyuserlocationdefs ^^J%
      (define-location-class \string"#1\string"^^J\space\space
        \space(#2))
    }%
    \edef\@xdyuserlocationnames{%
      \@xdyuserlocationnames^^J\space\space\space
      \string"#1\string"}%
  }
\fi
\renewcommand{\@do@wrglossary}[1]{%
\ifglxindy
  \expandafter\@glo@check@mkidxrangear\@glsnumberformat\@nil
  \def\@glo@range{}%
  \expandafter\if\@glo@prefix(\relax
    \def\@glo@range{:open-range}%
  \else
    \expandafter\if\@glo@prefix)\relax
    \def\@glo@range{:close-range}%
  \fi
  \fi
  \protected@edef\@glslocref{\theglentrycounter}%
  \@gls@checkmkidchars\@glslocref
  \glossary[\csname glo@#1@type\endcsname]{%
    (indexentry :tkey (\csname glo@#1@index\endcsname)
      :locref \string"\@glslocref\string" %
      :attr \string"\@glo@suffix\string" \@glo@range
    )
  }%
\else
  \@set@glo@numformat\@glo@numfmt\@gls@counter\@glsnumberformat
  \glossary[\csname glo@#1@type\endcsname]{%
    \string\glossaryentry{\csname glo@#1@index\endcsname
      \@gls@encapchar\@glo@numfmt}{\theglentrycounter}}%
  \fi
}
\def\@set@glo@numformat#1#2#3{%
  \expandafter\@glo@check@mkidxrangear#3\@nil
  \protected@edef#1{%
    \@glo@prefix setentrycounter[]#{2}%
  \expandafter\string\csname\@glo@suffix\endcsname
  }%
  \@gls@checkmkidchars#1%
}

```

```

\ifglxindy
\def\writeist{%
  \openout\glswrite=\listfilename
  \write\glswrite{;; xindy style file created by the glossaries
    package in compatible-2.07 mode}%
  \write\glswrite{;; for document '\jobname' on
    \the\year-\the\month-\the\day}%
  \write\glswrite{^^J; required styles^^J}
  \@for\@xdystyle:=\@xdyrequiredstyles\do{%
    \ifx\@xdystyle\@empty
    \else
    \protected@write\glswrite{{(require
      \string"\@xdystyle.xdy\string")}}%
    \fi
  }%
  \write\glswrite{^^J%
    ; list of allowed attributes (number formats)^^J}%
  \write\glswrite{(define-attributes ((\@xdyattributes)))}%
  \write\glswrite{^^J; user defined alphabets^^J}%
  \write\glswrite{\@xdyuseralphabets}%
  \write\glswrite{^^J; location class definitions^^J}%
  \protected@edef\@gls@roman{\@roman{0}\string"
    \string"roman-numbers-lowercase\string" :sep \string"}}%
  \@onelevel@sanitize\@gls@roman
  \edef\@tmp{\string" \string"roman-numbers-lowercase\string"
    :sep \string"}%
  \@onelevel@sanitize\@tmp
  \ifx\@tmp\@gls@roman
    \write\glswrite{(define-location-class
      \string"roman-page-numbers\string"^^J\space\space\space
      (\string"roman-numbers-lowercase\string")
      :min-range-length \@glsminrange)}%
  \else
    \write\glswrite{(define-location-class
      \string"roman-page-numbers\string"^^J\space\space\space
      (:sep "\@gls@roman")
      :min-range-length \@glsminrange)}%
  \fi
  \write\glswrite{(define-location-class
    \string"Roman-page-numbers\string"^^J\space\space\space
    (\string"roman-numbers-uppercase\string")
    :min-range-length \@glsminrange)}%
  \write\glswrite{(define-location-class
    \string"arabic-page-numbers\string"^^J\space\space\space
    (\string"arabic-numbers\string")
    :min-range-length \@glsminrange)}%
  \write\glswrite{(define-location-class
    \string"alpha-page-numbers\string"^^J\space\space\space
    (\string"alpha\string")
    :min-range-length \@glsminrange)}%

```

```

\write\glswrite{(define-location-class
  \string"Alpha-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"Appendix-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string"
  :sep \string"\@glsAlphacompositor\string"
  \string"arabic-numbers\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"arabic-section-numbers\string"^^J\space\space\space
  (\string"arabic-numbers\string"
  :sep \string"\glscompositor\string"
  \string"arabic-numbers\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\glsseeformat\string"
  :close \string"{}\string")}%
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
  :open \string"\string
  \glossarysection[\string\glossarytoctitle]{\string
  \glossarytitle}\string\glossary preamble\string~n\string\begin
  {theglossary}\string\glossaryheader\string~n\string" ^^J\space
  \space\space:close \string"\expandafter@gobble
  \string%\string~n\string
  \end{theglossary}\string\glossarypostamble
  \string~n\string" ^^J\space\space\space
  :tree)}}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\string~n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \string~n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\gls closebrace\string")}%
\write\glswrite{(markup-locref-list

```

```

:sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
:sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\string~n\string")}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
:open-head \string"\string\glsgroupheading
\glsopenbrace\string"^^J\space\space\space
:close-head \string"\glsclosebrace\string")}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%
\noist}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}
\def\writeist{\relax
\openout\glswrite=\istfilename
\write\glswrite{\expandafter\@gobble\string\% makeindex style file
created by the glossaries package}
\write\glswrite{\expandafter\@gobble\string\% for document
'\jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\@gls@actualchar'}
\write\glswrite{encap '\@gls@encapchar'}
\write\glswrite{level '\@gls@levelchar'}
\write\glswrite{quote '\@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
\glossarytoctitle]{\string\glossarytitle}\string
\glossarypreamble\string\n\string\begin{theglossary}\string
\glossaryheader\string\n\string"}

```

```

\write\glswrite{postamble \string"\string%\string\n\string
  \end{theglossary}\string\glossarypostamble\string\n
  \string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
  \string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}
\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
  \string"\string\relax \string\glsresetentrylist\string\n
  \string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
  \string"\string\relax \string\glsresetentrylist\string\n
  \string"}
\write\glswrite{delim_0 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_2 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_t \string"\string\}\string}\string"}
\write\glswrite{delim_n \string"\string\delimN \string"}
\write\glswrite{delim_r \string"\string\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
  \string"\string}\string\relax
  \string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"glscompositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF@empty
\else
  \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\noist
}
\fi
\renewcommand*{\noist}{\let\writeist\relax}

```

8.22 Rollback v4.49 (glossaries-compatible-307-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-307}[2021/11/01 v4.49 (NLCT)]
\newcommand{\compatglossarystyle}[2]{%
  \ifcsundef{@glscompstyle@#1}%
  {%
    \csdef{@glscompstyle@#1}{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary compatibility style ‘#1’ is already defined}{}%
  }%
}
\compatglossarystyle{inline}{%
  \renewcommand{\glossaryentryfield}[5]{%
    \glsinlinedopostchild
    \gls@inlinesep
    \def\glo@desc{##3}%
    \def\@no@post@desc{\nopostdesc}%
    \glsentryitem{##1}\glsinlinenameformat{##1}{##2}%
    \ifx\glo@desc\@no@post@desc
      \glsinlineemptydescformat{##4}{##5}%
    \else
      \ifstrempy{##3}%
        {\glsinlineemptydescformat{##4}{##5}}%
        {\glsinlinedescformat{##3}{##4}{##5}}%
      \fi
    \ifglshaschildren{##1}%
    {%
      \glsresetsubentrycounter
      \glsinlineparentchildseparator
      \def\gls@inlinesubsep{}%
      \def\gls@inlinepostchild{\glsinlinepostchild}%
    }%
    {}%
    \def\gls@inlinesep{\glsinlineseparator}%
  }%
  \renewcommand{\glossarysubentryfield}[6]{%
    \gls@inlinesubsep%
    \glsinlinesubnameformat{##2}{##3}%
    \glssubentryitem{##2}\glsinlinesubdescformat{##4}{##5}{##6}%
    \def\gls@inlinesubsep{\glsinlinesubseparator}%
  }%
}
\compatglossarystyle{list}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \item[\glsentryitem{##1}\glstarget{##1}{##2}]

```



```

        ##3\glspostdescription\space ##5}%
    \renewcommand*\glossarysubentryfield}[6]{%
        \glsentryitem{##2}%
        \glstarget{##2}{\strut}##4\glspostdescription\space ##6.}%
    }
\compatglossarystyle{listgroup}{%
    \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{listhypergroup}{%
    \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{altlist}{%
    \renewcommand*\glossaryentryfield}[5]{%
        \item[\glsentryitem{##1}\glstarget{##1}{##2}]%
        \mbox{}\par\nobreak\@afterheading
        ##3\glspostdescription\space ##5}%
    \renewcommand*\glossarysubentryfield}[6]{%
        \par
        \glsentryitem{##2}%
        \glstarget{##2}{\strut}##4\glspostdescription\space ##6}%
    }%
\compatglossarystyle{altlistgroup}{%
    \csuse{@glscompstyle@altlist}%
}%
\compatglossarystyle{altlisthypergroup}{%
    \csuse{@glscompstyle@altlist}%
}%
\compatglossarystyle{listdotted}{%
    \renewcommand*\glossaryentryfield}[5]{%
        \item[\makebox[\glslistdottedwidth][l]{%
            \glsentryitem{##1}\glstarget{##1}{##2}%
            \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##3}%
    \renewcommand*\glossarysubentryfield}[6]{%
        \item[\makebox[\glslistdottedwidth][l]{%
            \glsentryitem{##2}%
            \glstarget{##2}{##3}%
            \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##4}%
    }%
\compatglossarystyle{sublistdotted}{%
    \csuse{@glscompstyle@listdotted}%
    \renewcommand*\glossaryentryfield}[5]{%
        \item[\glsentryitem{##1}\glstarget{##1}{##2}]%
    }%
\compatglossarystyle{long}{%
    \renewcommand*\glossaryentryfield}[5]{%
        \glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
    \renewcommand*\glossarysubentryfield}[6]{%
        &
        \glsentryitem{##2}%
        \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
    }%

```

```

}%
\compatglossarystyle{longborder}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheader}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheaderborder}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{long3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \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]{%
    \glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
}%
\compatglossarystyle{long4colheader}{%
  \csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{long4colborder}{%
  \csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{long4colheaderborder}{%
  \csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4col}{%
  \csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colheader}{%
  \csuse{@glscompstyle@long4col}%
}%

```

```

\compatglossarystyle{altlong4colborder}{%
  \csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colheaderborder}{%
  \csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{longragged}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glstarget{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
    \tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6%
    \tabularnewline}%
}%
\compatglossarystyle{longraggedborder}{%
  \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longraggedheader}{%
  \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longraggedheaderborder}{%
  \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longragged3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glstarget{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{longragged3colborder}{%
  \csuse{@glscompstyle@longragged3col}%
}%
\compatglossarystyle{longragged3colheader}{%
  \csuse{@glscompstyle@longragged3col}%
}%
\compatglossarystyle{longragged3colheaderborder}{%
  \csuse{@glscompstyle@longragged3col}%
}%
\compatglossarystyle{altlongragged4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glstarget{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%

```

```

\compatglossarystyle{altlongragged4colheader}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colborder}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colheaderborder}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{index}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item\glsentryitem{##1}\textbf{\glstarget{##1}{##2}}%
    \ifx\relax##4\relax
      \else
        \space{##4}%
      \fi
    \space ##3\glspostdescription \space ##5}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    \ifcase##1\relax
      % level 0
      \item
    \or
      % level 1
      \subitem
      \glssubentryitem{##2}%
    \else
      % all other levels
      \subsubitem
    \fi
    \textbf{\glstarget{##2}{##3}}%
    \ifx\relax##5\relax
      \else
        \space{##5}%
      \fi
    \space##4\glspostdescription\space ##6}%
}%
\compatglossarystyle{indexgroup}{%
  \csuse{@glscompstyle@index}%
}%
\compatglossarystyle{indexhypergroup}{%
  \csuse{@glscompstyle@index}%
}%
\compatglossarystyle{tree}{%
  \renewcommand{\glossaryentryfield}[5]{%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glsentryitem{##1}\textbf{\glstarget{##1}{##2}}%
    \ifx\relax##4\relax
      \else
        \space{##4}%
      \fi
    \fi
  }%

```

```

\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\textbf{\glstarget{##2}{##3}}%
\ifx\relax##5\relax
\else
\space(##5)%
\fi
\space##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treegroup}{%
\csuse{@glscmpstyle@tree}%
}%
\compatglossarystyle{treehypergroup}{%
\csuse{@glscmpstyle@tree}%
}%
\compatglossarystyle{treenoname}{%
\renewcommand{\glossaryentryfield}[5]{%
\hangindent0pt\relax
\parindent0pt\relax
\glssubentryitem{##1}\textbf{\glstarget{##1}{##2}}%
\ifx\relax##4\relax
\else
\space(##4)%
\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\glstarget{##2}{\strut}%
##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treenonamegroup}{%
\csuse{@glscmpstyle@treenoname}%
}%
\compatglossarystyle{treenonamehypergroup}{%
\csuse{@glscmpstyle@treenoname}%
}%
\compatglossarystyle{almtree}{%
\renewcommand{\glossaryentryfield}[5]{%
\ifnum\@gls@prevlevel=0\relax
\else

```

```

        \settowidth{\glstreeindent}{\textbf{\@glswidestname\space}}%
        \hangindent\glstreeindent
        \parindent\glstreeindent
    \fi
    \makebox[0pt][r]{\makebox[\glstreeindent][l]{%
        \glstentryitem{##1}\textbf{\glstarget{##1}{##2}}}%
    \ifx\relax##4\relax
    \else
        (##4)\space
    \fi
    ##3\glspostdescription \space ##5\par
    \def\@gls@prevlevel{0}%
}%
\renewcommand{\glossarysubentryfield}[6]{%
    \ifnum##1=1\relax
        \glssubentryitem{##2}%
    \fi
    \ifnum\@gls@prevlevel=##1\relax
    \else
        \@ifundefined{@glswidestname\romannumeral##1}{%
            \settowidth{\gls@tmplen}{\textbf{\@glswidestname\space}}{%
            \settowidth{\gls@tmplen}{\textbf{%
                \csname @glswidestname\romannumeral##1\endcsname\space}}}%
        \ifnum\@gls@prevlevel<##1\relax
            \setlength\glstreeindent\gls@tmplen
            \addtolength\glstreeindent\parindent
            \parindent\glstreeindent
        \else
            \@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
                \settowidth{\glstreeindent}{\textbf{%
                    \@glswidestname\space}}}%
                \settowidth{\glstreeindent}{\textbf{%
                    \csname @glswidestname\romannumeral\@gls@prevlevel
                    \endcsname\space}}}%
            \addtolength\parindent{-\glstreeindent}%
            \setlength\glstreeindent\parindent
        \fi
    \fi
    \hangindent\glstreeindent
    \makebox[0pt][r]{\makebox[\gls@tmplen][l]{%
        \textbf{\glstarget{##2}{##3}}}%
    \ifx##5\relax\relax
    \else
        (##5)\space
    \fi
    ##4\glspostdescription\space ##6\par
    \def\@gls@prevlevel{##1}%
}%
}%
\compatglossarystyle{almtreeegroup}{%

```

```

\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{almtreehypergroup}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolindex}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexgroup}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexhypergroup}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcoltree}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{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]{%
\glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4\glspostdescription\space ##6%
}

```

```

        \tabularnewline}%
}%
\compatglossarystyle{superraggedborder}{%
  \csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheader}{%
  \csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheaderborder}{%
  \csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superragged3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{superragged3colborder}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheader}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheaderborder}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{altsuperragged4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%
\compatglossarystyle{altsuperragged4colheader}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colborder}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colheaderborder}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{super}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &

```



```

        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}##4\glspostdescription\space ##6\}%
}%
\compatglossarystyle{superborder}{%
  \csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{superheader}{%
  \csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{superheaderborder}{%
  \csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{super3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glssubentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\}%
}%
\compatglossarystyle{super3colborder}{%
  \csuse{@glscmpstyle@super3col}%
}%
\compatglossarystyle{super3colheader}{%
  \csuse{@glscmpstyle@super3col}%
}%
\compatglossarystyle{super3colheaderborder}{%
  \csuse{@glscmpstyle@super3col}%
}%
\compatglossarystyle{super4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glssubentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\}%
}%
\compatglossarystyle{super4colheader}{%
  \csuse{@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]{%
\ifcsemtyp{glo@glsdetoklabel{#1}@prefix}%
{#3}%
{#2}%
}
\newcommand*{\ifglshasprefixplural}[3]{%
\ifcsemtyp{glo@glsdetoklabel{#1}@prefixplural}%
{#3}%

```

```

    {#2}%
}
\newcommand*{\ifglshasprefixfirst}[3]{%
  \ifcempty{glo@glstdetoklabel{#1}@prefixfirst}%
  {#3}%
  {#2}%
}
\newcommand*{\ifglshasprefixfirstplural}[3]{%
  \ifcempty{glo@glstdetoklabel{#1}@prefixfirstplural}%
  {#3}%
  {#2}%
}
\providecommand{\@glsprefix@record@hook}[2]{%
  \ifdef\@glsextr@record
  {\@glsextr@record{#1}{#2}{glslink}}%
  {}%
}
\newcommand{\glsprefixsep}{}
\newrobustcmd{\pgls}{\@glshyp@opt\@pgls}
\newcommand*{\@pgls}[2][{}]{%
  \new@ifnextchar[%
  {\@pgls@{#1}{#2}}%
  {\@pgls@{#1}{#2}[]}%
}
\def\@pgls@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glstdoifexists{#2}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefix{#2}{\glsentryprefix{#2}\glsprefixsep}{%
      }%
      {%
        \ifglshasprefixfirst{#2}{\glsentryprefixfirst{#2}\glsprefixsep}{%
        }%
        \@glss@{#1}{#2}[#3]%
      }%
    }%
  }
}
\newrobustcmd{\pglsp1}{\@glshyp@opt\@pglsp1}
\newcommand*{\@pglsp1}[2][{}]{%
  \new@ifnextchar[%
  {\@pglsp1@{#1}{#2}}%
  {\@pglsp1@{#1}{#2}[]}%
}
\def\@pglsp1@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glstdoifexists{#2}%
  {%
    \ifglused{#2}%
    {%

```

```

        \ifglshasprefixplural{#2}{\glstentryprefixplural{#2}\glsprefixsep}{}%
    }%
    {%
        \ifglshasprefixfirstplural{#2}%
            {\glstentryprefixfirstplural{#2}\glsprefixsep}{}%
    }%
    \@glspl@{#1}{#2}[#3]%
}%
}
\newrobustcmd{\PglS}{\@gls@hyp@opt\PglS}
\newcommand*{\@PglS}[2][{}]{%
    \new@ifnextchar[%
        {\@PglS@{#1}{#2}}%
        {\@PglS@{#1}{#2}[{}]}%
    }
\def\@PglS@#1#2[#3]{%
    \@glsprefix@record@hook{#1}{#2}%
    \glsdoifexists{#2}%
    {%
        \ifglsused{#2}%
        {%
            \ifglshasprefix{#2}%
            {%
                \Glsentryprefix{#2}%
                \glsprefixsep
                \@gls@{#1}{#2}[#3]%
            }%
            {\@Gls@{#1}{#2}[#3]}%
        }%
        {%
            \ifglshasprefixfirst{#2}%
            {%
                \Glsentryprefixfirst{#2}%
                \glsprefixsep
                \@gls@{#1}{#2}[#3]%
            }%
            {\@Gls@{#1}{#2}[#3]}%
        }%
    }%
}
\newrobustcmd{\PglSpl}{\@gls@hyp@opt\PglSpl}
\newcommand*{\@PglSpl}[2][{}]{%
    \new@ifnextchar[%
        {\@PglSpl@{#1}{#2}}%
        {\@PglSpl@{#1}{#2}[{}]}%
    }
\def\@PglSpl@#1#2[#3]{%
    \@glsprefix@record@hook{#1}{#2}%
    \glsdoifexists{#2}%
    {%

```

```

\ifglused{#2}%
{%
  \ifglshasprefixplural{#2}%
  {%
    \Glentryprefixplural{#2}%
    \glsprefixsep
    \@glsp1@{#1}{#2}[#3]%
  }%
  {\@Glsp1@{#1}{#2}[#3]}%
}%
{%
  \ifglshasprefixfirstplural{#2}%
  {%
    \Glentryprefixfirstplural{#2}%
    \glsprefixsep
    \@glsp1@{#1}{#2}[#3]%
  }%
  {\@Glsp1@{#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}%
  {%
    \ifglschasprefixplural{#2}%
    {\mfirstucMakeUppercase{\glsentryprefixplural{#2}\glsprefixsep}}{%
  }%
  {%
    \ifglschasprefixfirstplural{#2}%
    {\mfirstucMakeUppercase{\glsentryprefixfirstplural{#2}\glsprefixsep}}{%
  }%
  \@GLSp1@{#1}{#2}{#3}%
}%
}

```

8.25 Rollback v4.49 (glossary-hypernav-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-hypernav}[2021/11/01 v4.49 (NLCT)]
\newcommand*\glsnavhyperlink}[3][\@glo@type]{%
  \protected@edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%
  \glslink{\glsnavhyperlinkname{#1}{#2}}{#3}}
\newcommand*\glsnavhyperlinkname}[2]{\glsn:#1@#2}
\newcommand*\glsnavhypertarget}[3][\@glo@type]{%
  \glsnavhypertarget{#1}{#2}{#3}%
}
\newcommand*\@glsnavhypertarget}[3]{%
  \protected@write\@auxout{}\string\@gls@hypergroup{#1}{#2}}%
  \@gls@target{\glsnavhyperlinkname{#1}{#2}}{#3}%
  \expandafter\let
  \expandafter\@gls@list\csname @gls@hypergroup@#1\endcsname
  \@for\@gls@elem:=\@gls@list\do{%
    \ifthenelse{\equal{\@gls@elem}{#2}}{\@endfortrue}{}}%
  \if@endfor
  \else
  \GlossariesWarningNoLine{Navigation panel
    for glossary type ‘#1’^^Jmissing group ‘#2’}%
  \gdef\gls@hypergroup@rerun{%
    \GlossariesWarningNoLine{Navigation panel
      has changed. Rerun LaTeX}}%
  \fi
}
\let\gls@hypergroup@rerun\relax
\AtEndDocument{\gls@hypergroup@rerun}
\newcommand*\@gls@hypergroup}[2]{%
  \@ifundefined{\@gls@hypergroup@list@#1}{%
    \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{#2}}%
  }%

```

```

\expandafter\let\expandafter\@gls@tmp
  \csname @gls@hypergroupplist@#1\endcsname
\expandafter\xdef\csname @gls@hypergroupplist@#1\endcsname{%
  \@gls@tmp,#2}%
}%
}
\newcommand*{\glsnavigation}{%
  \def\@gls@between{}%
  \ifcsundef{@gls@hypergroupplist@\@glo@type}%
  {%
    \def\@gls@list{}%
  }%
  {%
    \expandafter\let\expandafter\@gls@list
      \csname @gls@hypergroupplist@\@glo@type\endcsname
  }%
  \@for\@gls@tmp:=\@gls@list\do{%
    \@gls@between
    \@gls@getgrouptitle{\@gls@tmp}{\@gls@grptitle}%
    \glsnavhyperlink{\@gls@tmp}{\@gls@grptitle}%
    \let\@gls@between\glshypernavsep
  }%
}
\newcommand*{\glshypernavsep}{\space\textbar\space}
\newcommand*{\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*{\glsgrupheading}[1]{}%
  \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}}%
}%
\ifglschildren{##1}%
{%
  \glsresetsubentrycounter
  \glsinlineparentchildseparator
  \def\gls@inlinesubsep{}%
  \def\gls@inlinepostchild{\glsinlinepostchild}%
}%
{}%
\def\gls@inlinesep{\glsinlineseparator}%
}%
\renewcommand{\subglossentry}[3]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{%
    \glossentryname{##2}}%
  \gls@inlinesubdescformat{\glossentrydesc{##2}}{\glossentrysymbol{##2}}{##3}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%
\renewcommand*\{gls@groupskip}{}%
}
\newcommand*\{glsinlinedopostchild}{%
  \gls@inlinepostchild
  \def\gls@inlinepostchild{}%
}
\newcommand*\{glsinlineseparator}{;\space}
\newcommand*\{glsinlinesubseparator}{,\space}
\newcommand*\{glsinlineparentchildseparator}{:\space}
\newcommand*\{glsinlinepostchild}{\}
\newcommand*\{gls@postinline}{\gls@postdescription\space}
\newcommand*\{glsinlinenameformat}[2]{\glstarget{#1}{#2}}
\newcommand*\{glsinlinedescformat}[3]{\space#1}
\newcommand*\{glsinlineemptydescformat}[2]{}

```

```

\newcommand*\glsinlinesubnameformat}[2]{\glstarget{#1}{}}
\newcommand*\glsinlinesubdescformat}[3]{#1}

```

8.27 Rollback v4.49 (glossary-list-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossary-list}[2021/11/01 v4.49 (NLCT)]
\providecommand*\indexspace{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*\glslistgroupheaderfmt}[1]{#1}
\newcommand*\glslistnavigationitem}[1]{\item[#1]}
\newcommand*\glslistinit{%
  \ifdef\GetTitleStringDisableCommands
  {%
    \GetTitleStringSetup{expand}%
    \GetTitleStringDisableCommands{%
      \let\glsentryitem\@gobble
      \let\glstarget\@secondoftwo
      \let\glossentryname\glslistexpandedname
      \let\glslistgroupheaderfmt\@firstofone
      \let\glsgetgrouptitle\@firstofone
      \let\glsnavhypertarget\@secondoftwo
      \let\glsnavigation\relax
    }%
  }%
}
\newcommand{\glslistexpandedname}[1]{%
  \glsunexpandedfieldvalue{#1}{name}%
}
\newglossarystyle{list}{%
  \renewenvironment{theglossary}%
    {\glslistinit\begin{description}}{\end{description}}%
  \renewcommand*\{glossaryheader}{}%
  \renewcommand*\{glsgroupheading}[1]{}%
  \renewcommand*\{glossentry}[2]{%
    \item[\glsentryitem{##1}%
      \glstarget{##1}{\glossentryname{##1}}]
      \glossentrydesc{##1}\glspostdescription\space ##2}%
  \renewcommand*\{subglossentry}[3]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\space
    \glossentrydesc{##2}\glspostdescription\space ##3.}%
  \renewcommand*\{glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{listgroup}{%
  \setglossarystyle{list}%
}

```

```

\renewcommand*\glsgroupheading}[1]{%
  \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
\newglossarystyle{listhypergroup}{%
  \setglossarystyle{list}%
  \renewcommand*\glossaryheader}{%
    \glslistnavigationitem{\glsnavigation}}%
  \renewcommand*\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}]}]}
\newglossarystyle{altlist}{%
  \setglossarystyle{list}%
  \renewcommand*\glossentry}[2]{%
    \item[\glsentryitem{##1}%
      \glstarget{##1}{\glossentryname{##1}}%
      \mbox{}\par\nobreak\@afterheading
      \glossentrydesc{##1}\glspostdescription\space ##2}%
  \renewcommand*\subglossentry}[3]{%
    \par
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space ##3}%
}
\newglossarystyle{altlistgroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
\newglossarystyle{altlisthypergroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*\glossaryheader}{%
    \glslistnavigationitem{\glsnavigation}}%
  \renewcommand*\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}]}]}
\newglossarystyle{listdotted}{%
  \setglossarystyle{list}%
  \renewcommand*\glossentry}[2]{%
    \item[]\makebox[\glslistdottedwidth][l]{%
      \glsentryitem{##1}%
      \glstarget{##1}{\glossentryname{##1}}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##1}}%
  \renewcommand*\subglossentry}[3]{%
    \item[]\makebox[\glslistdottedwidth][l]{%
      \glssubentryitem{##2}%
      \glstarget{##2}{\glossentryname{##2}}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##2}}%
}
\newlength\glslistdottedwidth
\setlength{\glslistdottedwidth}{.5\hsize}
\newglossarystyle{sublistdotted}{%
  \setglossarystyle{listdotted}%
  \renewcommand*\glossentry}[2]{%

```

```

    \item[\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}}]{%
}

```

8.28 Rollback v4.49 (glossary-long-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-long}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{long}{%
  \renewenvironment{theglossary}{%
    {\begin{longtable}[lp{glsdescwidth}]}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
  \fi
}
\newglossarystyle{longborder}{%
  \setglossarystyle{long}%
  \renewenvironment{theglossary}{%
    \begin{longtable}[|lp{glsdescwidth}|]{\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{longheader}{%
  \setglossarystyle{long}%
  \renewcommand*{\glossaryheader}{%
    \bfseries \entryname & \bfseries \descriptionname\tabularnewline\endhead}%
}

```

```

\newglossarystyle{longheaderborder}{%
  \setglossarystyle{longborder}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries \entryname & \bfseries
    \descriptionname\tabularnewline\hline
    \endhead
    \hline\endfoot}%
}
\newglossarystyle{long3col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsdescwidth}p{\glspagelistwidth}}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{long3colborder}{%
  \setglossarystyle{long3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long3colheader}{%
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long3colheaderborder}{%
  \setglossarystyle{long3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}

```

```

}
\newglossarystyle{long4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{long4colheader}{%
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long4colborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long4colheaderborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}
}

```

```

\newglossarystyle{altlong4col}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheader}{%
  \setglossarystyle{long4colheader}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colborder}{%
  \setglossarystyle{long4colborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheaderborder}{%
  \setglossarystyle{long4colheaderborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%
    {\end{longtable}}%
}
}

```

8.29 Rollback v4.49 (glossary-longbooktabs-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-longbooktabs}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{booktabs}
\RequirePackage{glossary-long}[=v4.46]
\RequirePackage{glossary-longragged}[=v4.46]
\newglossarystyle{long-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname & \bfseries
    \descriptionname\tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{long3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%

```

```

        \toprule \bfseries \entryname &
        \bfseries \descriptionname &
        \bfseries \pagelistname
        \tabularnewline\midrule\endhead
        \bottomrule\endfoot}%
\ifglsnogroupskip
  \renewcommand*\{glsgroupskip}\{}%
\else
  \renewcommand*\{glsgroupskip}\{glspenaltygroupskip}%
\fi
}
\newglossarystyle{long4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long4col}%
  \renewcommand*\{glossaryheader}{%
    \toprule \bfseries \entryname &
    \bfseries \descriptionname &
    \bfseries \symbolname &
    \bfseries \pagelistname
    \tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsnogroupskip
    \renewcommand*\{glsgroupskip}\{}%
  \else
    \renewcommand*\{glsgroupskip}\{glspenaltygroupskip}%
  \fi
}
\newglossarystyle{altlong4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long4col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}\lp{\glsdescwidth}\lp{\glspagelistwidth}}%
    {\end{longtable}}%
}
\newglossarystyle{longragged-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}\l>{\raggedright}p{\glsdescwidth}}%
    {\end{longtable}}%
}
\newglossarystyle{longragged3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}\l>{\raggedright}p{\glsdescwidth}%
    >{\raggedright}p{\glspagelistwidth}}%
    {\end{longtable}}%
}
\newglossarystyle{altlongragged4col-booktabs}{%

```



```

\glspatchLToutput
\setglossarystyle{altlong4col-booktabs}%
\renewenvironment{theglossary}%
  {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
    >{\raggedright}p{\glspagelistwidth}}}%
  {\end{longtable}}%
}
\newcommand*{\glsLTpenaltycheck}{%
\ifnum\outputpenalty=-50\vskip-\normalbaselineskip\relax\fi
}
\newcommand{\glspenaltygroupskip}{%
\noalign{\penalty-50\vskip\normalbaselineskip}}
\let\@gls@org@LT@output\LT@output
\newcommand*{\glsrestoreLToutput}{\let\LT@output\@gls@org@LT@output}
\newcommand*{\glspatchLToutput}{%
\renewcommand*{\LT@output}{%
\ifnum\outputpenalty <-\@Mi
\ifnum\outputpenalty > -\LT@end@pen
\LT@err{floats and marginpars not allowed in a longtable}\@ehc
\else
\setbox\z@\vbox{\unvbox\@cclv}%
\ifdim \ht\LT@lastfoot>\ht\LT@foot
\dimen@\pagegoal
\advance\dimen@-\ht\LT@lastfoot
\ifdim\dimen@<\ht\z@
\setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%
\@makecol
\@outputpage
\setbox\z@\vbox{\box\LT@head\glsLTpenaltycheck}%
\fi
\fi
\global\@colroom\@colht
\global\vsizel\@colht
{\unvbox\z@\box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%
\fi
\else
\setbox\@cclv\vbox{\unvbox\@cclv\copy\LT@foot\vss}%
\@makecol
\@outputpage
\global\vsizel\@colroom
\copy\LT@head
\glsLTpenaltycheck
\nobreak
\fi
}%
}

```

8.30 Rollback v4.49 (glossary-longragged-2021-11-01.sty)

Version 4.49 preserved for rollback.

```
\ProvidesPackage{glossary-longragged}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{array}
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{longragged}{%
  \renewenvironment{theglossary}{%
    {\begin{longtable}[1>{\raggedright}p{glsdescwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{%}
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2%
    \tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}%
    \glspostdescription\space ##3%
    \tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
  \fi
}
\newglossarystyle{longraggedborder}{%
  \setglossarystyle{longragged}%
  \renewenvironment{theglossary}{%
    \begin{longtable}{|l|>{\raggedright}p{glsdescwidth}|}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{longraggedheader}{%
  \setglossarystyle{longragged}%
  \renewcommand*{\glossaryheader}{%
    \bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\endhead}%
}
```

```

}
\newglossarystyle{longraggedheaderborder}{%
  \setglossarystyle{longraggedborder}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\hline
    \endhead
    \hline\endfoot}%
}
\newglossarystyle{longragged3col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}[1>{\raggedright}p{\glsdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{longragged3colborder}{%
  \setglossarystyle{longragged3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}[|1|>{\raggedright}p{\glsdescwidth}|%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{longragged3colheader}{%
  \setglossarystyle{longragged3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{longragged3colheaderborder}{%
  \setglossarystyle{longragged3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline

```

```

        \bfseries\entryname&\bfseries\descriptionname&
        \bfseries\pagelistname\tabularnewline\hline\endhead
        \hline\endfoot}%
    }
\newglossarystyle{altlongragged4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}[l]>\raggedright}p{\glsdescwidth}l%
    >\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{altlongragged4colheader}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}[l]>\raggedright}p{\glsdescwidth}l%
    >\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{altlongragged4colborder}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>\raggedright}p{\glsdescwidth}l|l%
    >\raggedright}p{\glspagelistwidth}l}}}%
    {\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*{\gls mcols}{2}
\newglossarystyle{mcolindex}{%
  \setglossarystyle{index}%
  \renewenvironment{theglossary}{%
    {%
      \begin{multicols}{\gls mcols}
      \setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%
      \let\item\glstreeitem
      \let\subitem\glstreesubitem
      \let\subsubitem\glstreesubsubitem
    }%
    {\end{multicols}}%
  }
\newglossarystyle{mcolindexgroup}{%
  \setglossarystyle{mcolindex}%
  \renewcommand*{\gls groupheading}[1]{%
    \item\glstreegroupheaderfmt{\gls getgroup title{##1}}\indexspace}%
  }
\newglossarystyle{mcolindexhypergroup}{%
  \setglossarystyle{mcolindex}%
  \renewcommand*{\glossaryheader}{%
    \item\glstreenavigationfmt{\gls navigation}\indexspace}%
  \renewcommand*{\gls groupheading}[1]{%
    \item\glstreegroupheaderfmt
      {\gls navhypertarget{##1}}{\gls getgroup title{##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{mcoltreename}{%
  \setglossarystyle{treename}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}
\newglossarystyle{mcoltreenamegroup}{%
  \setglossarystyle{mcoltreename}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreenamehypergroup}{%
  \setglossarystyle{mcoltreename}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcoltreenamepannav}{%
  \setglossarystyle{treename}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcolalmtree}{%
  \setglossarystyle{almtree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
}
\newglossarystyle{mcolalmtreegroup}{%
  \setglossarystyle{mcolalmtree}%

```

```

\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcolalmtreehypergroup}{%
\setglossarystyle{mcolalmtree}%
\renewcommand*\glossaryheader{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcolalmtreespannav}{%
\setglossarystyle{almtree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsncols}[\noindent\glstreenavigationfmt{\glsnavigation}]
\def\@gls@prevlevel{-1}%
\mbox{}\par
}%
{\par\end{multicols}}%
\renewcommand*\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}

```

8.32 Rollback v4.49 (glossary-super-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-super}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
\newlength\glsdescwidth
\setlength{glsdescwidth}{0.6\hsize}
}{}

```



```

\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{super}{%
  \renewenvironment{theglossary}%
    {\tablehead{}}\tabletail{}%
    \begin{supertabular}{lp{glstdescwidth}}%
    {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}{|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\hline}%
    \tabletail{\hline}
    \begin{supertabular}{|lp{glstdescwidth}|}%

```

```

        {\end{supertabular}}%
    }
\newglossarystyle{super3col}{%
  \renewenvironment{theglossary}%
    {\tablehead{}}\tabletail{}}%
    \begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}%
    {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
  \fi
}
\newglossarystyle{super3colborder}{%
  \setglossarystyle{super3col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}}%
    \begin{supertabular}{|lp{\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}{|lp{\glsdescwidth}|p{\glspagelistwidth}|}%
    {\end{supertabular}}%
}
}

```

```

\newglossarystyle{super4col}{%
  \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}}%
    {\begin{supertabular}{1111}}{%
      \end{supertabular}}%
  \renewcommand*\glossaryheader{}%
  \renewcommand*\glsgroupheading}[1]{}%
  \renewcommand\glossentry[2]{%
    \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} & ##2\tabularnewline
  }%
  \renewcommand\subglossentry[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*\glsgroupskip{}%
  \else
    \renewcommand*\glsgroupskip}{& & \tabularnewline}%
  \fi
}
\newglossarystyle{super4colheader}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}%
    \tabletail{}}%
    {\begin{supertabular}{1111}}%
    {\end{supertabular}}%
}
\newglossarystyle{super4colborder}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}{|1|1|1|1|}%
    \end{supertabular}}%
}
\newglossarystyle{super4colheaderborder}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}{|1|1|1|1|}%
    \end{supertabular}}%
}

```

```

}
\newglossarystyle{altsuper4col}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{ }\tabletail{ }%
     \begin{supertabular}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuper4colheader}{%
  \setglossarystyle{super4colheader}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
     \bfseries\symbolname &
     \bfseries\pagelistname\tabularnewline}\tabletail{ }%
     \begin{supertabular}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuper4colborder}{%
  \setglossarystyle{super4colborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
     \begin{supertabular}%
       {llp{\glsdescwidth}llp{\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}%
       {llp{\glsdescwidth}llp{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
}

```

8.33 Rollback v4.49 (glossary-superragged-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-superragged}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{array}
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{ }

```

```

\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{superragged}{%
  \renewenvironment{theglossary}%
    {\tablehead{} \tabletail{}}%
    {\begin{supertabular}{1>{\raggedright}p{\glsdescwidth}}}%
    {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2%
    \tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3%
    \tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superraggedborder}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline} \tabletail{\hline}%
    \begin{supertabular}{||1>{\raggedright}p{\glsdescwidth}||}}%
    {\end{supertabular}}%
}
\newglossarystyle{superraggedheader}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries \entryname & \bfseries \descriptionname
    \tabularnewline}%
    \tabletail{}}%
    {\begin{supertabular}{1>{\raggedright}p{\glsdescwidth}}}%
    {\end{supertabular}}%
}
\newglossarystyle{superraggedheaderborder}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline\bfseries \entryname &
    \bfseries \descriptionname\tabularnewline\hline}%
}

```

```

        \tabletail{\hline}
        \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}||}%
        {\end{supertabular}}%
}
\newglossarystyle{superragged3col}{%
\renewenvironment{theglossary}%
{\tablehead{\tabletail}{%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}%
>{\raggedright}p{\glspagelistwidth}}}%
{\end{supertabular}}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1} &
##2\tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2} &
##3\tabularnewline
}%
\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi
}
\newglossarystyle{superragged3colborder}{%
\setglossarystyle{superragged3col}%
\renewenvironment{theglossary}%
{\tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}||%
>{\raggedright}p{\glspagelistwidth}||}%
{\end{supertabular}}%
}
\newglossarystyle{superragged3colheader}{%
\setglossarystyle{superragged3col}%
\renewenvironment{theglossary}%
{\tablehead{\bfseries\entryname&\bfseries\descriptionname&
\bfseries\pagelistname\tabularnewline}\tabletail}{%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}%
>{\raggedright}p{\glspagelistwidth}}}%
{\end{supertabular}}%
}
\newglossarystyle{superragged3colheaderborder}{%
\setglossarystyle{superragged3colborder}%
\renewenvironment{theglossary}%
{\tablehead{\hline

```

```

        \bfseries\entryname&\bfseries\descriptionname&
        \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
        >{\raggedright}p{\glspagelistwidth}|}%
    \end{supertabular}}%
}
\newglossarystyle{altsuperragged4col}{%
    \renewenvironment{theglossary}%
        {\tablehead{} \tabletail{}%
            \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}l%
                >{\raggedright}p{\glspagelistwidth}|}%
            \end{supertabular}}%
        \renewcommand*{\glossaryheader}{}%
        \renewcommand*{\glsgroupheading}[1]{}%
        \renewcommand{\glossentry}[2]{%
            \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
            \glossentrydesc{##1} &
            \glossentrysymbol{##1} & ##2\tabularnewline
        }%
        \renewcommand{\subglossentry}[3]{%
            &
            \glssubentryitem{##2}%
            \glstarget{##2}{\strut}\glossentrydesc{##2} &
            \glossentrysymbol{##2} & ##3\tabularnewline
        }%
        \ifglsnogroupskip
            \renewcommand*{\glsgroupskip}{}%
        \else
            \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
        \fi
    }
\newglossarystyle{altsuperragged4colheader}{%
    \setglossarystyle{altsuperragged4col}%
    \renewenvironment{theglossary}%
        {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
            \bfseries\symbolname &
            \bfseries\pagelistname\tabularnewline}\tabletail{}%
            \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}l%
                >{\raggedright}p{\glspagelistwidth}|}%
            \end{supertabular}}%
        }
\newglossarystyle{altsuperragged4colborder}{%
    \setglossarystyle{altsuper4col}%
    \renewenvironment{theglossary}%
        {\tablehead{\hline}\tabletail{\hline}%
            \begin{supertabular}%
                { |l|>{\raggedright}p{\glsdescwidth}||l|%
                    >{\raggedright}p{\glspagelistwidth}|}%
            \end{supertabular}}%

```

```

}
\newglossarystyle{altsuperragged4colheaderborder}{%
  \setglossarystyle{altsuperragged4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname &
      \bfseries\descriptionname &
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}%
      \ll>{\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{\glsgetgrouptitle{##1}}%
        \indexspace
    }%
}
\newglossarystyle{indexhypergroup}{%
    \setglossarystyle{index}%
    \renewcommand*{\glossaryheader}{%
        \item\glstreenavigationfmt{\glsnavigation}\indexspace}%
    \renewcommand*{\glsgroupheading}[1]{%
        \item\glstreegroupheaderfmt
            {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
        \indexspace}%
}
\newglossarystyle{tree}{%
    \renewenvironment{theglossary}%
        {\setlength{\parindent}{0pt}%
         \setlength{\parskip}{0pt plus 0.3pt}}%
        {}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
        \hangindent0pt\relax
        \parindent0pt\relax
        \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
        \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
        \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
    }%
}

```

```

}%
\renewcommand{\subglossentry}[3]{%
  \hangindent##1\glstreeindent\relax
  \parindent##1\glstreeindent\relax
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
  \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
  \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3\par
}%
\renewcommand*{\glsgroupskip}{\ifglsgnogroupskip\else\indexspace\fi}}
\newglossarystyle{treegroup}{%
  \setglossarystyle{tree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par
    \indexspace}%
}
\newglossarystyle{treehypergroup}{%
  \setglossarystyle{tree}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsgroupnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt
    {\glsgrouphypertarget{##1}{\glsgrouptitle{##1}}}\par
    \indexspace}%
}
\newlength\glstreeindent
\setlength{\glstreeindent}{10pt}
\newglossarystyle{treenoname}{%
  \renewenvironment{theglossary}%
    {\setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}}%
    {}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glssubentryitem{##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}{\ifglsgnogroupskip\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{\glsgnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt
{\glsgnnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}
\newrobustcmd*{\glsgfindwidesttoplevelname}[1][\@glo@types]{%
\dimen@=0pt\relax
\gls@tmplen=0pt\relax
\forallglossaries[#1]{\@gls@type}%
{%
\forallglsentries[\@gls@type]{\@glo@label}%
{%
\ifglshasparent{\@glo@label}%
}%
{%
\settowidth{\dimen@}%
{\glstreenamefmt{\glsentryname{\@glo@label}}}%
\ifdim\dimen@>\gls@tmplen
\gls@tmplen=\dimen@
\letcs{\@glswidestname}{glo\@glsdetoklabel{\@glo@label}@name}%
\fi
}%
}%
}
\newcommand*{\glssetwidest}[2][0]{%
\expandafter\def\csname @glswidestname\romannumeral#1\endcsname{%
#2}%
}
\newcommand*{\@glswidestname}{}
\newcommand*{\glstreenamebox}[2]{%
\makebox[#1][1]{#2}%
}
\newglossarystyle{almtree}{%

```

```

\renewenvironment{theglossary}%
  {\def\@gls@prevlevel{-1}%
   \mbox{}\par}%
  {\par}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
  \ifnum\@gls@prevlevel=0\relax
  \else
    \settowidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%
  \fi
  \hangindent\glstreeindent
  \parindent\glstreeindent
  \makebox[Opt][r]{\glstreenamebox{\glstreeindent}{%
    \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}}}%
  \ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}%
  \glossentrydesc{##1}\glspostdescription \space ##2\par
  \def\@gls@prevlevel{0}%
}%
\renewcommand{\subglossentry}[3]{%
  \ifnum##1=1\relax
  \glssubentryitem{##2}%
  \fi
  \ifnum\@gls@prevlevel=##1\relax
  \else
    \ifundefined{@glswidestname\romannumeral##1}{%
      \settowidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}{%
        \settowidth{\gls@tmplen}{\glstreenamefmt{%
          \csname @glswidestname\romannumeral##1\endcsname\space}}}%
      \ifnum\@gls@prevlevel<##1\relax
        \setlength\glstreeindent\gls@tmplen
        \addtolength\glstreeindent\parindent
        \parindent\glstreeindent
      \else
        \ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
          \settowidth{\glstreeindent}{\glstreenamefmt{%
            \@glswidestname\space}}{%
            \settowidth{\glstreeindent}{\glstreenamefmt{%
              \csname @glswidestname\romannumeral\@gls@prevlevel
                \endcsname\space}}}%
          \addtolength\parindent{-\glstreeindent}%
          \setlength\glstreeindent\parindent
        \fi
        \fi
      \fi
    \fi
  \hangindent\glstreeindent
  \makebox[Opt][r]{\glstreenamebox{\gls@tmplen}{%
    \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
  \ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
  \glossentrydesc{##2}\glspostdescription\space ##3\par
  \def\@gls@prevlevel{##1}%

```

```

}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{almtreegroup}{%
\setglossarystyle{almtree}%
\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
\par\indexspace}%
}
\newglossarystyle{almtreehypergroup}{%
\setglossarystyle{almtree}%
\renewcommand*{\glossaryheader}{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}}

```

8.35 Rollback v4.52 (glossaries-2022-11-03.sty)

```

\NeedsTeXFormat{LaTeX2e}
\DeclareRelease{v4.46}{2020-03-19}{glossaries-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-2021-11-01.sty}
\DeclareCurrentRelease{v4.52}{2022-11-03}
\ProvidesPackage{glossaries}[2022/11/03 v4.52 (NLCT)]
\RequirePackage{ifthen}
\RequirePackage{xkeyval}[2006/11/18]
\RequirePackage{mfirstuc}
\newcommand{\glsentencecase}[1]{\makefirstuc{#1}}
\ExplSyntaxOn
\newcommand{\glslowercase}[1]{\text_lowercase:n{#1}}
\newcommand{\glsuppercase}[1]{\text_uppercase:n{#1}}
\ifdef\MFUsentencecase
{%
\newcommand{\glsmfuexcl}[1]{\MFUexcl{#1}}
\newcommand{\glsmfublocker}[1]{\MFUblocker{#1}}
\newcommand{\glsmfuaddmap}[2]{\MFUaddmap{#1}{#2}}
}
{

```

```

\RequirePackage{textcase}
\providecommand{\MFUsentencecase}[1]{\text_titlecase_first:n{#1}}
\providecommand{\glsmfuexcl}[1]{
  \tl_if_in:NnF \l_text_case_exclude_arg_tl {#1}
  {
    \tl_put_right:Nn \l_text_case_exclude_arg_tl {#1}
  }
}
\renewcommand*{\mfirstucMakeUppercase}{\MakeTextUppercase}
\newcommand{\glsmfublocker}[1]{\glsmfuexcl{#1}}
\newcommand{\glsmfuaddmap}[2]{\glsmfuexcl{#1}\glsmfublocker{#2}}
}
\ExplSyntaxOff
\RequirePackage{xfor}
\RequirePackage{datatool-base}
\RequirePackage{amsgen}
\RequirePackage{etoolbox}
\newif\if@gls@docloaded
\@ifpackageloaded{doc}{%
  {%
    \@gls@docloadedtrue
  }%
  {%
    \@ifclassloaded{nlctdoc}{\@gls@docloadedtrue}{\@gls@docloadedfalse}%
  }
}
\if@gls@docloaded
  \let\glsorg@theglossary\theglossary
  \let\glsorg@endtheglossary\endtheglossary
  \let\glsorg@PrintChanges\PrintChanges
  \renewcommand{\PrintChanges}{%
    \begingroup
      \let\theglossary\glsorg@theglossary
      \let\endtheglossary\glsorg@endtheglossary
      \glsorg@PrintChanges
    \endgroup
  }
\fi
\newif\if@gls@debug
\define@choicekey{glossaries.sty}{debug}[\gls@debug@val\gls@debug@nr]{%
{true,false,showtargets,showaccsupp}[true]{%
  \ifcase\gls@debug@nr\relax
    % debug=true
    \@gls@debugtrue
    \renewcommand*{\GlossariesWarning}[1]{%
      \PackageWarning{glossaries}{##1}%
    }%
    \renewcommand*{\GlossariesWarningNoLine}[1]{%
      \PackageWarningNoLine{glossaries}{##1}%
    }%
  \let\@gls@showtarget\@gobble

```

```

\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\or
% debug=false
\@gls@debugfalse
\let\glsshowtarget\@gobble
\let\glsshowaccsupp\@gobblethree
\PackageInfo{glossaries}{debug mode OFF}%
\or
% debug=showtargets
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowtarget}{\@glsshowtarget}%
\or
% debug=showaccsupp
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
\fi
}
\newcommand*\glsshowtarget}[1]{%
\ifmmode
\glsshowtargetinner{#1}%
\else
\ifinner
\glsshowtargetinner{#1}%
\else
\glsshowtargetouter{#1}%
\fi
\fi
}
\newcommand*\glsshowtargetfonttext}[1]{%
\ifmmode \nfss@text{\glsshowtargetfont #1}\else {\glsshowtargetfont #1}\fi
}
\newcommand*\glsshowtargetinner}[1]{\glsshowtargetfonttext{[#1]}}
\newcommand*\glsshowtargetouter}[1]{%
\glsshowtargetsymbol\marginpar{\glsshowtargetsymbol\glsshowtargetfont #1}}
\newcommand*\glsshowtargetsymbol}{\tiny$\triangleright$}
\newcommand*\glsshowtargetfont}{\ttfamily\footnotesize}

```

```

\newcommand*\@glsshowtarget}[1]{}
\newcommand*\glsdoshowtarget}[2]{\@glsshowtarget{#1}#2}
\newrobustcmd*\@@glsshowtarget}[1]{%
  \begingroup
  \protected@edef\@gls@tmp{#1}%
  \@onelevel@sanitize\@gls@tmp
  \expandafter\glsshowtarget\expandafter{\@gls@tmp}%
  \endgroup
}
\newcommand*\@glsshowaccsupp}[3]{}
\newrobustcmd*\@glsshowaccsupp}[3]{%
  \ifstrempy{#1}%
  {\@@glsshowtarget{/#2 (#3)}}%
  {\@@glsshowtarget{/#2 (#3) [#1]}}%
}
\newcommand*\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries\space (or move
  \string\newglossaryentry\space
  definitions into the preamble)}%
  {You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key. It may
  be that the 'see' key has been written to the .glsdefs
  file from the previous run, in which case you need to
  move your definitions
  to the preamble if you don't want to use
  \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoinde}x{%
  [\gls@seenoinde@val\gls@seenoinde@nr]{error,warn,ignore}{%
    \ifcase\gls@seenoinde@nr
      \renewcommand*\@gls@see@noindex}{%
        \PackageError{glossaries}%
        {'\gls@xr@key' key may only be used after \string\makeglossaries\space
        or \string\makenoidxglossaries}%
        {You must use \string\makeglossaries\space
        or \string\makenoidxglossaries\space before defining
        any entries that have a '\gls@xr@key' key}%
      }%
    \or
      \renewcommand*\@gls@see@noindex}{%
        \GlossariesWarning{'\gls@xr@key' key ignored}%
      }%
    \or
      \renewcommand*\@gls@see@noindex}{}%
    \fi
  }
}

```



```

\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}%
  {\newcommand*\@@glossarysec}{section}}%
  {\newcommand*\@@glossarysec}{chapter}}
\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
subsection,subsubsection,paragraph,subparagraph}[section]{%
  \renewcommand*\@@glossarysec}{#1}}
\newcommand*\@@glossarysecstar}{*}%
\newcommand*\@@glossaryseclabel}{%
\newcommand*\glsautoprefix}{%
\define@choicekey{glossaries.sty}{numberedsection}%
  [\gls@numberedsection@val\gls@numberedsection@nr]{%
false,nolabel,autolabel,nameref}[nolabel]{%
  \ifcase\gls@numberedsection@nr\relax
    \renewcommand*\@@glossarysecstar}{*}%
    \renewcommand*\@@glossaryseclabel}{%
  \or
    \renewcommand*\@@glossarysecstar}{%
    \renewcommand*\@@glossaryseclabel}{%
  \or
    \renewcommand*\@@glossarysecstar}{%
    \renewcommand*\@@glossaryseclabel}{%
    \label{\glsautoprefix@glo@type}}%
  \or
    \renewcommand*\@@glossarysecstar}{*}%
    \renewcommand*\@@glossaryseclabel}{%
    \protected@edef\@currentlabelname{\glossarytoctitle}%
    \label{\glsautoprefix@glo@type}}%
  \fi
}
\@ifpackageloaded{classicthesis}
{\newcommand*\@glossary@default@style}{index}}
{\newcommand*\@glossary@default@style}{list}}
\define@key{glossaries.sty}{style}{%
  \def\@glossary@default@style{#1}}
}
\newcommand*\@gls@declareoption}[2]{%
  \DeclareOptionX{#1}{#2}%
  \DeclareOption{#1}{#2}}
}
\newcommand*\glossaryentrynumbers}[1]{#1\gls@save@numberlist{#1}}
\@gls@declareoption{nonumberlist}{%
  \renewcommand*\glossaryentrynumbers}[1]{\gls@save@numberlist{#1}}%
}
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{}
\gls@savenumberlistfalse
\newcommand*\@glo@seeautonumberlist{}
\@gls@declareoption{seeautonumberlist}{%
  \renewcommand*\@glo@seeautonumberlist}{%

```

```

\def\@glo@prefix{\glsnextpages}%
}%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{}
\glsesclocationsfalse
\newcommand*\@gls@loadlong{\RequirePackage{glossary-long}}
\@gls@declareoption{no long}{\renewcommand*\@gls@loadlong{}}
\IfFileExists{supertabular.sty}{%
\newcommand*\@gls@loadsuper{\RequirePackage{glossary-super}}}{%
\newcommand*\@gls@loadsuper{}}
\@ifpackage{loaded}{xtab}{\renewcommand*\@gls@loadsuper{}}{}
\@gls@declareoption{nosuper}{\renewcommand*\@gls@loadsuper{}}{}
\newcommand*\@gls@loadlist{\RequirePackage{glossary-list}}
\@gls@declareoption{no list}{%
\renewcommand*\@gls@loadlist}{%
\ifdefstring{\@glossary@default@style}{list}%
{\let\@glossary@default@style\relax}%
}%
}%
}
\newcommand*\@gls@loadtree{\RequirePackage{glossary-tree}}
\@gls@declareoption{no tree}{\renewcommand*\@gls@loadtree{}}
\@gls@declareoption{no styles}{%
\renewcommand*\@gls@loadlong}{%
\renewcommand*\@gls@loadsuper}{%
\renewcommand*\@gls@loadlist}{%
\renewcommand*\@gls@loadtree}{%
\let\@glossary@default@style\relax
}
\newcommand*\@gls@postdescription}{%
\ifglsnopostdot\else.\spacefactor\sfcode`. \fi
}
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{}
\glsnopostdotfalse
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{}
\glsnogroupskipfalse
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{}
\@ifclassloaded{memoir}
{%
\glsucmarktrue
}%
{%
\glsucmarkfalse
}
\newcommand*\@gls@define@glossary@entry@counter}{%
\ifglsentrycounter
\ifundef\c@glossaryentry
{%
\ifx\@gls@counterwithin\@empty
\newcounter{glossaryentry}%

```

```

        \else
            \newcounter{glossaryentry}[\@gls@counterwithin]%
        \fi
        \def\theHglossaryentry{\currentglossary.\theglossaryentry}%
    }%
    {}%
\fi
}
\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\glstentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
    \renewcommand*{\@gls@counterwithin}{#1}%
    \glstentrycountertrue
    \@gls@define@glossaryentrycounter
}
\newcommand*{\@gls@counterwithin}{}
\newcommand{\@gls@define@glossarysubentrycounter}{%
    \ifundef\c@glossarysubentry
    {%
        \ifglssubentrycounter
            \ifglstentrycounter
                \newcounter{glossarysubentry}[glossaryentry]%
            \else
                \newcounter{glossarysubentry}%
            \fi
        \fi
        \def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
    \fi
    }%
    {}%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
\glssubentrycounterfalse
\newcommand{\glsindexingsetting}{\ifglsxindy xindy\else makeindex\fi}
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none,clear}{%
    \renewcommand*{\@glo@default@sorttype}{#1}%
    \csname @gls@setupsort@#1\endcsname
}
\newcommand*{\glsprestandardsort}[3]{%
    \glsdosanitizesort
}
\newcommand*{\@glo@check@sortallowed}[1]{}
\newcommand*{\@gls@setupsort@standard}{%
    \def\do@glo@storeentry{\@glo@storeentry}%
    \def\@gls@defsortcount##1{%
    \def\@gls@defsort##1##2{%
        \ifx\@glo@sort\@glsdefaultsort
            \let\@glo@sort\@glo@name
        \fi
        \let\glsdosanitizesort\@gls@sanitizesort
    }
}

```

```

\glsprestandardsort{\@glo@sort}{##1}{##2}%
\expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\@gls@setupsort@standard
\newcommand*\glsortnumberfmt [1]{%
\ifnum#1<100000 0\fi
\ifnum#1<10000 0\fi
\ifnum#1<1000 0\fi
\ifnum#1<100 0\fi
\ifnum#1<10 0\fi
\number#1%
}
\newcommand*\@gls@setupsort@def}{%
\def\do@glo@storeentry{\@glo@storeentry}%
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\ifcsundef{glossary@##1@sortcount}%
{\@gls@defsortcount{##1}}%
{}}%
\expandafter\global\expandafter
\advance\csname glossary@##1@sortcount\endcsname by 1\relax
\expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
\expandafter\glsortnumberfmt
{\csname glossary@##1@sortcount\endcsname}}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*\@gls@setupsort@use}{%
\let\do@glo@storeentry\@gobble
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\expandafter\gdef\csname glo@##2@sort\endcsname{}%
}%
\def\@gls@setsort##1{%
\protected@edef\@glo@parent{\csname glo@##1@parent\endcsname}%
\ifx\@glo@parent\@empty
\else
\expandafter\@gls@setsort\expandafter{\@glo@parent}%
\fi
\protected@edef\@glo@type{\csname glo@##1@type\endcsname}%

```

```

\protected@edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
\ifx\@gls@tmp\@empty
\expandafter\global\expandafter
\advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
\expandafter\protected@xdef\csname glo@##1@sort\endcsname{%
\expandafter\glssortnumberfmt
{\csname glossary@\@glo@type @sortcount\endcsname}}%
\@glo@storeentry{##1}%
\fi
}%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*{\@gls@setupsort@none}{%
\def\glsindexingsetting{none}%
\def\do@glo@storeentry##1{%
\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
\expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort
}%
\def\@gls@setsort##1{%
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
{Option sort=none not allowed with \string##1}%
{(Use sort=def instead)}}%
}
\newcommand*{\@gls@setupsort@clear}{%
\def\glsindexingsetting{none}%
\def\do@glo@storeentry##1{%
\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{\csgdef{glo@##2@sort}{}}%
\def\@gls@setsort##1{%
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
{Option sort=clear not allowed with \string##1}%
{(Use sort=def instead)}}%
}
\newcommand*{\glsdefmain}{%
\if@gls@docloaded
\newglossary[glg2]{main}{gls2}{glo2}{\glossaryname}%
\else
\newglossary{main}{gls}{glo}{\glossaryname}%
\fi
\newcommand*{\gls@tr@set@main@toctitle}{%
\translatelet{\glossarytoctitle}{Glossary}%
}%
}
\newcommand*{\glsdefaulttype}{main}
\newcommand*{\acronymtype}{\glsdefaulttype}
\@gls@declareoption{nomain}{%
\let\glsdefaulttype\relax
\renewcommand*{\glsdefmain}{%
}

```

```

\define@boolkey{glossaries.sty}[gls]{acronym}[true]{%
  \ifglsacronym
    \renewcommand{\@gls@do@acronymsdef}{%
      \DeclareAcronymList{acronym}%
      \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
      \renewcommand*{\acronymtype}{acronym}%
      \newcommand*\gls@tr@set@acronym@toctitle}{%
        \translatelet{\glossarytoctitle}{Acronyms}%
      }%
    }%
  \else
    \let\@gls@do@acronymsdef\relax
  \fi
}
\AtBeginDocument{%
  \ifglsacronym
    \providecommand*\printacronyms[1][1]{%
      \printglossary[type=\acronymtype,#1]}%
    \fi
}
\newcommand*\@gls@do@acronymsdef{}
\@gls@declareoption{acronyms}{%
  \glsacronymtrue
  \def\@gls@do@acronymsdef{%
    \DeclareAcronymList{acronym}%
    \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
    \renewcommand*{\acronymtype}{acronym}%
    \newcommand*\gls@tr@set@acronym@toctitle}{%
      \translatelet{\glossarytoctitle}{Acronyms}%
    }%
  }%
}
\newcommand*\@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@gls@isacronymlist
\newcommand*\@gls@checkisacronymlist}[1]{%
\glsIfListOfAcronyms{#1}%
{\@gls@isacronymlisttrue}{\@gls@isacronymlistfalse}%
}
\newcommand*\@SetAcronymLists}[1]{%
\renewcommand*\@gls@acronymlists{#1}%
}
\define@key{glossaries.sty}{acronymlists}{%
\DeclareAcronymList{#1}%
}
\newcommand*\@gls@counter{page}
\define@key{glossaries.sty}{counter}{%
\renewcommand*\@gls@counter{#1}%
}
\newcommand*\@gls@nohyperlist{}
\newcommand*\@GlsDeclareNoHyperList}[1]{%
\ifdefempty\@gls@nohyperlist
{%
\renewcommand*\@gls@nohyperlist{#1}%
}%
{%
\appto\@gls@nohyperlist{,#1}%
}%
}
\define@key{glossaries.sty}{nohypertypes}{%
\GlsDeclareNoHyperList{#1}%
}
\newcommand*\@GlossariesWarning}[1]{%
\PackageWarning{glossaries}{#1}%
}
\newcommand*\@GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{#1}%
}
\newcommand*\@glosortentrieswarning{%
\typeout{Using TeX to sort glossary entries---this may
take a while}%
}
\@gls@declareoption{nowarn}{%
\if@gls@debug
\GlossariesWarning{Warnings can't be suppressed in debug mode}%
\else

```

```

\renewcommand*\GlossariesWarning}[1]{%
\renewcommand*\GlossariesWarningNoLine}[1]{}%
\renewcommand*\glosortentrieswarning}{}%
\renewcommand*\@gls@missinglang@warn}[2]{}%
\fi
}
\newcommand*\@gls@missinglang@warn}[2]{%
\PackageWarningNoLine{glossaries}%
{No language module detected for ‘#1’.\MessageBreak
Language modules need to be installed separately.\MessageBreak
Please check on CTAN for a bundle called\MessageBreak
‘glossaries-#2’ or similar}%
}
\@gls@declareoption{noLANGwarn}{%
\renewcommand*\@gls@missinglang@warn}[2]{}%
}
\newcommand*\@gls@warnonglossdefined}{%
\GlossariesWarning{Overriding \string\printglossary}%
}
\newcommand*\@gls@warnontheGlossdefined}{%
\GlossariesWarning{Overriding ‘theGlossary’ environment}%
}
\@gls@declareoption{noRedefwarn}{%
\renewcommand*\@gls@warnonglossdefined}{}%
\renewcommand*\@gls@warnontheGlossdefined}{}%
}
\newcommand*\glssetexpandfield}[1]{%
\csdef{gls@assign@#1@field}##1##2{%
\@gls@expand@field{##1}{#1}{##2}%
}%
}
\newcommand*\glssetnoexpandfield}[1]{%
\csdef{gls@assign@#1@field}##1##2{%
\@gls@noexpand@field{##1}{#1}{##2}%
}%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}
\glssetnoexpandfield{descplural}
\newcommand*\@gls@sanitizename}{%
\glssetnoexpandfield{name}
\newcommand*\@gls@sanitizesymbol}{%
\glssetnoexpandfield{symbol}
\glssetnoexpandfield{symbolplural}
\newcommand*\@gls@sanitizesort}{%
\ifgls@sanitizesort
\@gls@sanitizesort
\else
\@gls@nosanitizesort
\fi

```



```

}
\newcommand*{\@gls@sanitizesort{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@gls@nosanitizesort}{%}
\newcommand*{\@gls@noidx@sanitizesort{%
  \ifdefvoid\@glo@sort
  }%
  {%
  \expandafter\@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
  \def\@glo@sort{#1#2}%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@gls@noidx@nosanitizesort}{%
  \ifdefvoid\@glo@sort
  }%
  {%
  \expandafter\@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
  \bgroup
  \glsnoidxstripaccents
  \protected@xdef\@glo@sort{#1#2}%
  \egroup
  \let\@glo@sort\@glo@sort
}
\newcommand*\glsnoidxstripaccents{%
  \let\IeC\@firstofone
  \let\add@accent@\@secondoftwo
  \let\@text@composite@x\@secondoftwo
  \let\@tabacckludge\@secondoftwo
  \expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
  \let'\@firstofone
  \let'\@firstofone
  \let^\@firstofone
  \let"@@firstofone
  \let\u\@firstofone
  \let\t\@firstofone
  \let\d\@firstofone
  \let\r\@firstofone
  \let=\@firstofone
  \let.\@firstofone
  \let~\@firstofone

```

```

\let\v\@firstofone
\let\H\@firstofone
\let\c\@firstofone
\let\b\@firstofone
\let\a\@secondoftwo
\def\AE{AE}%
\def\ae{ae}%
\def\OE{OE}%
\def\oe{oe}%
\def\AA{AA}%
\def\aa{aa}%
\def\L{L}%
\def\l{l}%
\def\O{O}%
\def\o{o}%
\def\SS{SS}%
\def\ss{ss}%
\def\th{th}%
\def\TH{TH}%
\def\dh{dh}%
\def\DH{DH}%
}
\@ifl@t@r\fmtversion{2019/10/01}
{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}
\newcommand{\@gls@removedoption}[1]{%
\PackageError{glossaries}{obsolete package option ‘#1’ has been
removed. Rollback required or use a newer alternative}
{Try \string\usepackage[#1]{glossaries}[=v4.46]}%
}
\define@boolkey[gls]{sanitize}{description}[true]{%
\GlossariesWarning{sanitize={description} package option deprecated}%
\ifgls@sanitize@description
\glssetnoexpandfield{desc}%
\glssetnoexpandfield{descplural}%
\else
\glssetexpandfield{desc}%
\glssetexpandfield{descplural}%
\fi
}
\define@boolkey[gls]{sanitize}{name}[true]{%
\GlossariesWarning{sanitize={name} package option deprecated}%
\ifgls@sanitize@name
\glssetnoexpandfield{name}%
\else
\glssetexpandfield{name}%
\fi
}
}

```

```

\define@boolkey[glS]{sanitize}{symbol}[true]{%
  \GlossariesWarning{sanitize={symbol} package option deprecated}%
  \ifglS@sanitize@symbol
    \glSsetnoexpandfield{symbol}%
    \glSsetnoexpandfield{symbolplural}%
  \else
    \glSsetexpandfield{symbol}%
    \glSsetexpandfield{symbolplural}%
  \fi
}
\define@boolkey{glossaries.sty}[glS]{sanitizesort}[true]{%
  \ifglSsanitizesort
    \glSsetnoexpandfield{sortvalue}%
    \renewcommand*{\@glS@noidx@setsanitizesort}{%
      \glSsanitizesorttrue
      \glSsetnoexpandfield{sortvalue}%
    }%
  \else
    \glSsetexpandfield{sortvalue}%
    \renewcommand*{\@glS@noidx@setsanitizesort}{%
      \glSsanitizesortfalse
      \glSsetexpandfield{sortvalue}%
    }%
  \fi
}
\glSsanitizesorttrue
\glSsetnoexpandfield{sortvalue}%
\newcommand*{\@glS@noidx@setsanitizesort}{%
  \glSsanitizesortfalse
  \glSsetexpandfield{sortvalue}%
}
\define@choicekey[glS]{sanitize}{sort}{true,false}[true]{%
  \setbool{glSsanitizesort}{#1}%
  \ifglSsanitizesort
    \glSsetnoexpandfield{sortvalue}%
  \else
    \glSsetexpandfield{sortvalue}%
  \fi
  \GlossariesWarning{sanitize={sort} package option
    deprecated. Use sanitizesort instead}%
}
\define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
  \ifthenelse{\equal{#1}{none}}{%
    {%
      \GlossariesWarning{sanitize package option deprecated}%
      \glSsetexpandfield{name}%
      \glSsetexpandfield{symbol}%
      \glSsetexpandfield{symbolplural}%
      \glSsetexpandfield{desc}%
      \glSsetexpandfield{descplural}%
    }
  }
}

```

```

}%
{%
  \setkeys[glS]{sanitizE}{#1}%
}%
}
\newif\ifglstranslate
\newcommand*\@glS@usetranslator{%
  \@ifpackageloaded{polyglossia}%
  {%
    \let\glSifusetranslator\@secondoftwo
  }%
  {%
    \@ifpackageloaded{babel}%
    {%
      \IfFileExists{translator.sty}%
      {%
        \RequirePackage{translator}%
        \let\glSifusetranslator\@firstoftwo
      }%
      {}%
    }%
    {}%
  }%
}
\newcommand{\glSifusedtranslatorDict}[3]{%
  \glSifusetranslator
  {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
  {#3}%
}
\@glS@declareoption{notranslate}{%
  \glstranslatefalse
  \let\@glS@usetranslator\relax
  \let\glSifusetranslator\@secondoftwo
}
\define@choicekey{glossaries.sty}{translate}{%
  [\glS@translate@val\glS@translate@nr]%
  {true,false,babel}[true]%
  {%
    \ifcase\glS@translate@nr\relax
    \glstranslatetrue
    \renewcommand*\@glS@usetranslator{%
      \@ifpackageloaded{polyglossia}%
      {%
        \let\glSifusetranslator\@secondoftwo
      }%
      {}%
    }%
    \@ifpackageloaded{babel}%
    {%
      \IfFileExists{translator.sty}%
      {%

```

```

        \RequirePackage{translator}%
        \let\glsifusetranslator\@firstoftwo
    }%
    {}%
} %
{}%
} %
} %
\or
\glstranslatefalse
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\or
\glstranslatetrue
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\fi
}
\glstranslatefalse
\let\glsifusetranslator\@secondoftwo
\@ifpackageloaded{translator}%
{%
\glstranslatetrue
\let\glsifusetranslator\@firstoftwo
}%
{%
\@for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
{
\@ifpackageloaded{\gls@thissty}%
{%
\glstranslatetrue
\endfortrue
}%
}%
}
}
\newcommand{\@gls@requested@languages}{}
\define@key{glossaries.sty}{languages}{%
\renewcommand{\@gls@requested@languages}{#1}%
\glstranslatetrue
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
}
\define@boolkey{glossaries.sty}[gls]{indexonlyfirst}[true]{}
\glsindexonlyfirstfalse
\define@boolkey{glossaries.sty}[gls]{hyperfirst}[true]{}
\glshyperfirsttrue
\newcommand*{\@gls@setacrstyle}{}
\define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{}
\ifglsacrfootnote

```

```

    \@gls@removedoption{footnote}%
  \fi
}
\define@boolkey{glossaries.sty}[glsacr]{description}[true]{%
  \ifglsacrdescription
    \@gls@removedoption{description}%
  \fi
}
\define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{%
  \ifglsacrsmallcaps
    \@gls@removedoption{smallcaps}%
  \fi
}
\define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{%
  \ifglsacrsmaller
    \@gls@removedoption{smaller}%
  \fi
}
\define@boolkey{glossaries.sty}[glsacr]{dua}[true]{%
  \ifglsacrdua
    \@gls@removedoption{dua}%
  \fi
}
\define@boolkey{glossaries.sty}[glsacr]{shortcuts}[true]{}
\newcommand*{\glsorder}{word}
\newcommand*{\@glsorder}[1]{}
\define@choicekey{glossaries.sty}{order}{word,letter}{%
  \def\glsorder{#1}}
\newif\ifglsxindy
\glsxindyfalse
\@gls@declareoption{makeindex}{\glsxindyfalse}
\define@boolkey[gls]{xindy}[glsnumbers][true]{}
\gls@xindy@glsnumberstrue
\def\@xdy@main@language{\languagename}%
\newcommand*{\gls@xdy@set@lang}[1]{%
  \ifcsundef{\@xdy@#1@language}%
    {%
      \ifcsundef{\@xdy@\glsdefaulttype @language}
        {\let\@gls@langmod\@xdy@main@language}%
        {\letcs\@gls@langmod{\@xdy@\glsdefaulttype @language}}}%
    }%
  {\letcs\@gls@langmod{\@xdy@#1@language}}%
}
\define@key[gls]{xindy}{language}{\def\@xdy@main@language{#1}}
\ifdefvoid\inputencodingname
  {\def\gls@codepage{utf8}}%
  {\def\gls@codepage{\inputencodingname}}
\define@key[gls]{xindy}{codepage}{\def\gls@codepage{#1}}
\define@key{glossaries.sty}{xindy}[][%
  \glsxindytrue

```

```

\setkeys[glS]{xindy}{#1}%
}
\@glS@declareoption{xindygloss}{%
  \glSxindytrue
}
\@glS@declareoption{xindynoglsnumbers}{%
  \glSxindytrue
  \glSxindy@glSnumbersfalse
}
\providecommand{\@domakeglossaries}[1]{\if@filesW #1\fi}
\@glS@declareoption{disablemakegloss}{%
  \ifdefequal\makeglossaries\@no@makeglossaries
  {%
    \GlossariesWarning{Option ‘disablemakegloss’ has no effect
      (\string\makenoidxglossaries\space has already been used)}%
  }%
  {%
    \ifdefequal\@makeglossary\@gobble
    {%
      \GlossariesWarning{Option ‘disablemakegloss’ has no effect
        (\string\makeglossaries\space has already been used)}%
    }%
  }%
  \renewcommand{\@domakeglossaries}[1]{%
    \PackageInfo{glossaries}{\string\makeglossaries\space and
      \string\makenoidxglossaries\space have been disabled}%
  }%
}
}
\@glS@declareoption{restoremakegloss}{%
  \ifdefequal\makeglossaries\@no@makeglossaries
  {%
    \GlossariesWarning{Option ‘restoremakegloss’ has no effect
      (\string\makenoidxglossaries\space has already been used)}%
  }%
  {%
    \ifdefequal\@makeglossary\@gobble
    {%
      \GlossariesWarning{Option ‘restoremakegloss’ has no effect
        (\string\makeglossaries\space has already been used)}%
    }%
  }%
  \PackageInfo{glossaries}{\string\makeglossaries\space and
    \string\makenoidxglossaries\space have been restored}%
  \let\@domakeglossaries\@firstofone
}
}
\newcommand*{\@do@write@glSlabels}{%

```

```

\AtEndDocument{\do@write@glslabels}%
\let\do@write@glslabels\relax
}
\newcommand*\do@write@glslabels{%
  \newwrite\@gls@labelsfile
  \immediate\openout\@gls@labelsfile=\jobname.glslabels
  \protected@edef\@gls@fulllist{\@glo@types}%
  \ifdefempty{\@ignored@glossaries}{}%
  {\protected@eappto\@gls@fulllist{\@ignored@glossaries}}%
  \forallglsentries[\@gls@fulllist]{\@glsentry}%
  {%
    \ifdefempty{\@glsentry}{}%
    {%
      \ifx\do@write@glslabelsname@gobble
        \def\@gls@thisentryname{%
          \else
            \letcs{\@gls@thisentryname}{glo@\@glsdetoklabel{\@glsentry}@name}%
            \@onelevel@sanitize\@gls@thisentryname
          \fi
          \immediate\write\@gls@labelsfile
            {\@glsentry\do@write@glslabelsname{\@gls@thisentryname}}%
        }%
      }%
    \immediate\closeout\@gls@labelsfile
  }
}
\newcommand*\do@write@glslabelsname}[1]{%
  \begingroup
  \catcode'\^^I12
  \gdef\@write@glslabelsname#1{\^^I#1}
  \endgroup
  \@gls@declareoption{writeglslabels}{\do@write@glslabels
    \let\do@write@glslabelsname@gobble}
  \@gls@declareoption{writeglslabelnames}{\do@write@glslabels
    \let\do@write@glslabelsname\@write@glslabelsname}
  \newif\ifglsautomake
  \newcommand{\gls@automake@nr}{1}
  \newcommand{\gls@automake@makegloss}{%
    \newcommand{\gls@automake@option}{automake=\gls@automake@val}
    \newcommand*\@gls@do@automake@err}{%
      \PackageError{glossaries}{You must use
        \string\makeglossaries\space with \gls@automake@option}
      {%
        Either remove the \gls@automake@option\space setting or
        add \string\makeglossaries\space to your document preamble.%
      }%
    }%
  }%
}
\define@choicekey{glossaries.sty}{automake}%
  [\gls@automake@val\gls@automake@nr]{true,false,immediate,makegloss,lite,delayed}[immediate]{%
  \ifcase\gls@automake@nr\relax
    \GlossariesWarning{automake=true deprecated. Use automake=delayed for a delayed write}%
  }%
}

```



```

\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomakefalse
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{makeglossaries}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
\or
\def\gls@automake@nr{0}%
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\fi
\renewcommand{\gls@automake@option}{automake=\gls@automake@val}%
\ifglsautomake
\let\@gls@doautomake\@gls@do@automake@err
\else
\renewcommand*\@gls@doautomake{}%
\fi
}
\glsautomakefalse
\newcommand*\@gls@doautomake{}
\AtEndDocument{\@gls@doautomake}
\@gls@declareoption{automakegloss}{%
\glsautomaketrue
\renewcommand{\gls@automake@option}{automakegloss}%
\def\gls@automake@val{makegloss}%
\def\gls@automake@nr{3}%
\renewcommand{\gls@automake@makegloss}{makeglossaries}%
\let\@gls@doautomake\@gls@do@automake@err
}
\@gls@declareoption{automakeglosslite}{%
\glsautomaketrue
\renewcommand{\gls@automake@option}{automakeglosslite}%
\def\gls@automake@val{lite}%
\def\gls@automake@nr{4}%
\renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
\let\@gls@doautomake\@gls@do@automake@err
}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
\ifgls savewrites
\renewcommand*\@gls@writefiles{\@gls@writefiles}%
\else
\let\@gls@writefiles\@empty

```

```

\fi
}
\glssavewritesfalse
\let\glswritefiles\@empty
\define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{%
  \ifbool{glscompatible-3.07}%
  {%
    \@gls@removedoption{compatible-3.07}%
  }%
  {}%
}
\boolfalse{glscompatible-3.07}
\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
  \ifbool{glscompatible-2.07}%
  {%
    \@gls@removedoption{compatible-2.07}%
  }%
  {}%
}
\boolfalse{glscompatible-2.07}
\let\gls@original@makeglossary\makeglossary
\let\gls@original@glossary\glossary
\newcommand*{\@gls@override@glossary}[1][main]{%
  \GlossariesWarning{Use of \string\glossary\space with
glossaries.sty is deprecated. \MessageBreak Indexing should be performed
with the user level \MessageBreak commands, such as \string\gls\space or
\string\glsadd. If you need the \MessageBreak original definition of
\string\glossary\space use the package \MessageBreak options
kernelglossredefs=false (to restore the \MessageBreak former definition of
\string\glossary) and nomain (if the \MessageBreak file extensions cause a
conflict)}}%
\gls@glossary{#1}%
}
\define@choicekey{glossaries.sty}{kernelglossredefs}{%
  [\gls@kernelglossredefs@val\gls@kernelglossredefs@nr]{true,false,nowarn}[true]{%
  {%
    \ifcase\gls@kernelglossredefs@nr\relax
    \def\glossary{\@gls@override@glossary}%
    \def\makeglossary{%
      \GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is deprecated. Use \string\makeglossaries\space
instead. If you need the original definition of
\string\makeglossary\space use the package options
kernelglossredefs=false (to prevent redefinition of
\string\makeglossary) and nomain (if the file extensions cause a
conflict)}}%
      \makeglossaries
    }%
  }%
  \or
  \let\glossary\gls@original@glossary

```

```

\let\makeglossary\gls@original@makeglossary
\or
\def\makeglossary{\makeglossaries}%
\renewcommand*{\@gls@override@glossary}[1][main]{%
  \gls@glossary{##1}%
}%
\fi
}
\@gls@declareoption{symbols}{%
  \let\@gls@do@symbolsdef\@gls@symbolsdef
}
\newcommand*{\@gls@do@symbolsdef}{%
\newcommand*{\@gls@symbolsdef}{%
  \newglossary[slg]{symbols}{sls}{slo}{\glsymbolsgroupname}%
  \newcommand*{\printsymbols}[1][\printglossary[type=symbols,##1]]{%
  \newcommand*{\gls@tr@set@symbols@toctitle}{%
    \translatelet{\glossarytoctitle}{Symbols (glossaries)}%
  }%
}%
\@gls@declareoption{numbers}{%
  \let\@gls@do@numbersdef\@gls@numbersdef
}
\newcommand*{\@gls@do@numbersdef}{%
\newcommand*{\@gls@numbersdef}{%
  \newglossary[nlg]{numbers}{nls}{nlo}{\glnumbersgroupname}%
  \newcommand*{\printnumbers}[1][\printglossary[type=numbers,##1]]{%
  \newcommand*{\gls@tr@set@numbers@toctitle}{%
    \translatelet{\glossarytoctitle}{Numbers (glossaries)}%
  }%
}%
\@gls@declareoption{index}{%
  \ifx\@gls@do@indexdef\@empty
  \let\@gls@do@indexdef\@gls@indexdef
  \fi
}
\@gls@declareoption{noglossaryindex}{%
  \let\@gls@do@indexdef\relax
}
\newcommand*{\@gls@do@indexdef}{%
\newcommand*{\@gls@indexdef}{%
  \newglossary[ilg]{index}{ind}{idx}{\indexname}%
  \newcommand*{\printindex}[1][\printglossary[type=index,##1]]{%
  \newcommand*{\newterm}[2][%
    \newglossaryentry{##2}%
    {type={index},name={##2},description={\nopostdesc},##1}}
  \let\@gls@do@indexdef\relax
}%
}
\define@choicekey{glossaries.sty}{mfirstuc}{%
  [\gls@mfirstuc@val\gls@mfirstuc@nr]%
  {expanded,unexpanded}%
}

```

```

{%
  \ifcase\gls@mfirstuc@nr
  % expanded
  \renewcommand*\glsmakefirstuc}[1]{\MFUsentencecase{#1}}%
  \or
  % unexpanded
  \renewcommand*\glsmakefirstuc}[1]{\MFUsentencecase{\unexpanded{#1}}}%
  \fi
}
\@for\CurrentOption :=\@declaredoptions\do{%
  \ifx\CurrentOption\@empty
  \else
  \@expandtwoargs
  \in@ {\, \CurrentOption ,}{, \@classoptionslist, \@curroptions,}%
  \ifin@
  \@use@ption
  \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
  \fi
  \fi
}
\ProcessOptionsX
\ifglsautomake
\ifdef\directlua{\RequirePackage{shellesc}}{}
\fi
\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,notranslate,languages,nolong,nosuper,notree,nostyles,%
nomain,noglossaryindex}
\newcommand*\setupglossaries}[1]{%
  \renewcommand*\@gls@setacrstyle}{}%
  \ifglsacrshortcuts
  \def\@gls@setupshortcuts{\glsacrshortcutstrue}%
  \else
  \def\@gls@setupshortcuts{%
    \ifglsacrshortcuts
    \DefineAcronymSynonyms
    \fi
  }%
  \fi
  \glsacrshortcutsfalse
  \let\@gls@do@numbersdef\relax
  \let\@gls@do@symbolssdef\relax
  \let\@gls@do@indexdef\relax
  \let\@gls@do@acronymsdef\relax
  \ifglsubentrycounter
  \let\@gls@doentrycounterdef\relax
  \else
  \let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
  \fi
  \ifglsubentrycounter

```

```

        \let\@gls@dosomeentrycounterdef\relax
    \else
        \let\@gls@dosomeentrycounterdef\@gls@define@glossarysubentrycounter
    \fi
    \setkeys{glossaries.sty}{#1}%
    \@gls@setacrstyle
    \@gls@setupshortcuts
    \@gls@do@acronymsdef
    \@gls@do@numbersdef
    \@gls@do@symbolssdef
    \@gls@do@indexdef
    \@gls@doentrycounterdef
    \@gls@dosomeentrycounterdef
}
\ifthenelse{\equal{\glscounter}{section}}%
{%
    \ifcsundef{chapter}{}%
    {%
        \let\@gls@old@chapter\@chapter
        \def\@chapter[#1]#2{\@gls@old@chapter[#1]}{#2}%
        \ifcsundef{hyperdef}{\hyperdef{section}{\thesection}}{}}%
    }%
}%
{}
\newcommand*{\@gls@onlypremakeg}{%
\newcommand*{\@onlypremakeg}[1]{%
    \ifx\@gls@onlypremakeg\@empty
        \def\@gls@onlypremakeg{#1}%
    \else
        \expandafter\toks@\expandafter{\@gls@onlypremakeg}%
        \edef\@gls@onlypremakeg{\the\toks@,\noexpand#1}%
    \fi
}
\newcommand*{\@disable@onlypremakeg}{%
\@for\@thiscs:=\@gls@onlypremakeg\do{%
    \expandafter\@disable@premakecs\@thiscs%
}}
\newcommand*{\@disable@premakecs}[1]{%
    \def#1{\PackageError{glossaries}{\string#1\space may only be
    used before \string\makeglossaries}{You can't use
    \string#1\space after \string\makeglossaries}}%
}
\providecommand*{\glossaryname}{Glossary}
\providecommand*{\acronymname}{Acronyms}
\newcommand*{\glssettoctitle}[1]{%
    \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}
\providecommand*{\entryname}{Notation}
\providecommand*{\descriptionname}{Description}
\providecommand*{\symbolname}{Symbol}
\providecommand*{\pagelistname}{Page List}

```

```

\providecommand*\glssymbolsgroupname}{Symbols}
\providecommand*\glsnumbersgroupname}{Numbers}
\newcommand*\glspluralsuffix}{s}
\newcommand*\glsacrpluralsuffix}{\glspluralsuffix}
\newcommand*\glsupacrpluralsuffix}{\glstextup{\glsacrpluralsuffix}}
\providecommand*\seename}{see}
\providecommand*\andname}{\&}
\newcommand*\RequireGlossariesLang}[1]{%
  \ifundefined{ver@glossaries-#1.ldf}{\input{glossaries-#1.ldf}}{}%
}
\newcommand*\ProvidesGlossariesLang}[1]{%
  \ProvidesFile{glossaries-#1.ldf}%
}
\newcommand*\addglossarytocaptions}[1]{}
\ifglstranslate
  \RequirePackage{tracklang}[2019/11/30]
  \ifvoid\@gls@requested@languages{}%
  {%
    \@for\@gls@thislang:=\@gls@requested@languages\do{%
      \TrackIfKnownLanguage{\@gls@thislang}{}%
      {\GlossariesWarning{unknown language ‘\@gls@thislang’}}%
    }%
  }%
  \@gls@usetranslator
  \@ifpackageloaded{translator}
  {%
    \ifboolexpr
    {
      test {\ifdefstring{\trans@languages}{English}}
      and not
      test {\ifdefstring{bbl@loaded}{english}}
    }
    {%
      \let\glsifusetranslator\@secondoftwo
    }%
    {%
      \usedictionary{glossaries-dictionary}%
      \renewcommand*\addglossarytocaptions}[1]{%
        \ifcsundef{captions#1}{}%
        {%
          \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
          \expandafter\toks@\expandafter{\@gls@tmp
            \renewcommand*\glossaryname}{\translate{Glossary}}%
          }%
          \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
        }%
      }%
    }%
  }%
}{}%

```

```

\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
  }%
}%
{}%
\glsifusetranslator
{%
  \renewcommand*\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*\glossaryname{\translate{Glossary}}%
  \renewcommand*\acronymname{\translate{Acronyms}}%
  \renewcommand*\entryname{\translate{Notation (glossaries)}}%
  \renewcommand*\descriptionname{%
    \translate{Description (glossaries)}}%
  \renewcommand*\symbolname{\translate{Symbol (glossaries)}}%
  \renewcommand*\pagelistname{%
    \translate{Page List (glossaries)}}%
  \renewcommand*\glssymbolsgroupname{%
    \translate{Symbols (glossaries)}}%
  \renewcommand*\glsnumbersgroupname{%
    \translate{Numbers (glossaries)}}%
  }{}%
\fi
\DeclareRobustCommand*\nopostdesc{}
\newcommand*\@nopostdesc{%
  \let\org@glspostdescription\glspostdescription
  \def\glspostdescription{%
    \let\glspostdescription\org@glspostdescription}%
}
\newcommand*\@no@post@desc{\nopostdesc}
\newcommand*\glspar{\par}
\newcommand*\setStyleFile[1]{%
  \renewcommand*\gls@istfilebase{#1}%
}

```

```

\ifglxindy
  \def\istfilename{\gls@istfilebase.xdy}
\else
  \def\istfilename{\gls@istfilebase.ist}
\fi
}
\@onlypremakeg\setStyleFile
\ifglxindy
  \def\istfilename{\gls@istfilebase.xdy}
\else
  \def\istfilename{\gls@istfilebase.ist}
\fi
\newcommand*\gls@istfilebase{\jobname}
\newcommand*\@istfilename[1]{
\newcommand*\glscompositor{.}
\newcommand*\glsSetCompositor[1]{%
  \renewcommand*\glscompositor{#1}}
\@onlypremakeg\glsSetCompositor
\newcommand*\@glsAlphaCompositor{\glscompositor}
\ifglxindy
  \newcommand*\glsSetAlphaCompositor[1]{%
    \renewcommand*\@glsAlphaCompositor{#1}}
\else
  \newcommand*\glsSetAlphaCompositor[1]{%
    \glsnoxywarning\glsSetAlphaCompositor}
\fi
\@onlypremakeg\glsSetAlphaCompositor
\newcommand*\gls@suffixF{}
\newcommand*\glsSetSuffixF[1]{%
  \renewcommand*\gls@suffixF{#1}}
\@onlypremakeg\glsSetSuffixF
\newcommand*\gls@suffixFF{}
\newcommand*\glsSetSuffixFF[1]{%
  \renewcommand*\gls@suffixFF{#1}}
}
\ifcsundef{hyperlink}%
{%
  \newcommand*\glsnumberformat[1]{#1}%
}%
{%
  \newcommand*\glsnumberformat[1]{\glshypernumber{#1}}%
}
\newcommand{\glslocationstoencap}[3]{\csuse{#2}{\csuse{#1}{#3}}}
\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*\glossarypreamble{%
  \csuse{@glossarypreamble@currentglossary}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaultttype]{%
  \ifglossaryexists*#1}%

```



```

{\csgdef{@glossary preamble@#1}{#2}}%
{\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*{\glossary postamble}{}
\newcommand*{\glossary section}[2][\@gls@title]{%
\def\@gls@title{#2}%
\ifcsundef{phantom section}%
{%
\@glossary section{#1}{#2}%
}%
{%
\@pglossary section{#1}{#2}%
}%
\glsglossary mark{\glossary toctitle}%
}
\ifcsundef{glossary mark}%
{%
\newcommand{\glsglossary mark}[1]{\glossary mark{#1}}
}%
{%
\@ifclassloaded{memoir}
{%
\newcommand{\glsglossary mark}[1]{%
\ifglsucmark
\markboth{\memUchead{#1}}{\memUchead{#1}}%
\else
\markboth{#1}{#1}%
\fi
}
}%
{%
\newcommand{\glsglossary mark}[1]{%
\ifglsucmark
\@mkboth{\glsupercase{#1}}{\glsupercase{#1}}%
\else
\@mkboth{#1}{#1}%
\fi
}
}
}
\providecommand{\glossary mark}[1]{%
\ifglsucmark
\@mkboth{\glsupercase{#1}}{\glsupercase{#1}}%
\else
\@mkboth{#1}{#1}%
\fi
}
\newcommand*{\setglossary section}[1]{%
\setkeys{glossaries.sty}{section=#1}}
\newcommand*{\@glossary section}[2]{%

```

```

\ifdefempty\@glossarysecstar
{%
  \csname\@glossarysec\endcsname[#1]{#2}%
}%
{%
  \csname\@glossarysec\endcsname*{#2}%
  \@gls@toc{#1}{\@glossarysec}%
}%
\@glossaryseclabel
}
\newcommand*\@pglossarysection}[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*\glsnoindeywarning}[1]{%
    \GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
}
\newcommand*\glsnomakeindexwarning}[1]{%
    \GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%
}
\ifglsxindy
    \edef\xdyattributes{\string"default\string"}%
\fi
\ifglsxindy
    \edef\xdyattributelist{}%
\fi
\ifglsxindy
    \def\xdylocref{}
\fi
\newcommand*\@gls@ifinlist}[4]{%
    \def\@do@ifinlist##1,#1,##2\end@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]{%
        \glsnoindeywarning\GlsAddXdyAttribute

```

```

}
\fi
\newcommand*\@disabled@glSaddxdycounters{%
  \PackageError{glossaries}{\string\GlsAddXdyCounters\space
    can't be used after \string\GlsAddXdyAttribute}{Move all
    occurrences of \string\GlsAddXdyCounters\space before the first
    instance of \string\GlsAddXdyAttribute}%
}
\ifglSxindy
  \newcommand*\@glSaddxdyattribute[2]{%
    \edef\@xdyattributes{\@xdyattributes ^^J
      \string"#2#1\string"}%
    \expandafter\toks@\expandafter{\@xdylocref}%
    \edef\@xdylocref{\the\toks@ ^^J%
      (markup-locref
      :open \string"\glStildechar n%
        \expandafter\string\csname glS#2X#1\endcsname
        \string" ^^J
      :close \string"\string" ^^J
      :attr \string"#2#1\string")}%
    \expandafter\gdef\csname glS#2X#1\endcsname##1##2{%
      \setentrycounter[##1][##2]\csname #1\endcsname{##2}%
    }%
  }
}
\newcommand*\GlsAddXdyAttribute[1]{%
  \ifx\@xdyattributelist\@empty
    \edef\@xdyattributelist{#1}%
  \else
    \edef\@xdyattributelist{\@xdyattributelist,#1}%
  \fi
  \edef\@xdyattributes{\@xdyattributes ^^J
    \string"#1\string"}%
  \@for\@this@counter:=\@xdycounters\do{%
    \protected@edef\glS@do@addxdyattribute{%
      \noexpand\@glSaddxdyattribute{#1}{\@this@counter}%
    }
    \glS@do@addxdyattribute
  }%
  \let\GlsAddXdyCounters\@disabled@glSaddxdycounters
}
\@onlypremakeg\GlsAddXdyAttribute
\else
  \newcommand*\GlsAddXdyAttribute[1]{%
    \glSnoxindywarning\GlsAddXdyAttribute}
\fi
\ifglSxindy
\newcommand*{\@glS@addpredefinedattributes}{%
  \GlsAddXdyAttribute{glSnumberformat}
  \GlsAddXdyAttribute{textrm}
  \GlsAddXdyAttribute{textsf}
}

```

```

\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glshypernumber}
\GlsAddXdyAttribute{hyperrm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hypertt}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}
\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}
\GlsAddXdyAttribute{glsignore}
}
\else
\let\@gls@addpredefinedattributes\relax
\fi
\def\@xdyuseralphabets{}
\ifglxindy
\newcommand*{\GlsAddXdyAlphabet}[2]{%
\edef\@xdyuseralphabets{%
\@xdyuseralphabets ~^J
(define-alphabet "#1" (#2))}}
\else
\newcommand*{\GlsAddXdyAlphabet}[2]{%
\glsnoxindywarning\GlsAddXdyAlphabet}
\fi
\ifglxindy
\newcommand*{\@gls@xdy@locationlist}{%
roman-page-numbers,%
Roman-page-numbers,%
arabic-page-numbers,%
alpha-page-numbers,%
Alpha-page-numbers,%
Appendix-page-numbers,%
arabic-section-numbers%
}
\protected@edef\@gls@roman{\@roman{0}\string"
\string"roman-numbers-lowercase\string" :sep \string"}%
\@onelevel@sanitize\@gls@roman
\edef\@gls@tmp{\string" \string"roman-numbers-lowercase\string"
:sep \string"}%
\@onelevel@sanitize\@gls@tmp

```

```

\ifx\@gls@tmp\@gls@roman
  \expandafter
  \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
    \string"roman-numbers-lowercase\string"%
  }%
\else
  \expandafter
  \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{
    :sep \string"\@gls@roman\string"%
  }%
\fi
\expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
  \string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
  \string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
  \string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
  \string"ALPHA\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
  \string"ALPHA\string"
  :sep \string"\@gls@Alphacompositor\string"
  \string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
  \string"arabic-numbers\string"
  :sep \string"\glscompositor\string"
  \string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglxindy
  \newcommand\@xdycrossrefhook{}
\fi
\ifglxindy
  \newcommand*\GlsAddXdyLocation}[3][[]]{%
    \def\@gls@tmp[#1]{%
      \ifx\@gls@tmp\@empty
        \edef\@xdyuserlocationdefs{%
          \@xdyuserlocationdefs ^^J%
          (define-location-class \string"#2\string"^^J\space\space
            \space(:sep \string"{}\glsopenbrace\string" #3
              :sep \string"\glsclosebrace\string"))
        }%
      \else

```

```

\edef\@xdyuserlocationdefs{%
  \@xdyuserlocationdefs ^^J%
  (define-location-class \string"#2\string"^^J\space\space
  \space(:sep "\glsoopenbrace"
    #1
    :sep "\glsclosebrace\glsoopenbrace" #3
    :sep "\glsclosebrace"))
}%
\fi
\edef\@xdyuserlocationnames{%
  \@xdyuserlocationnames^^J\space\space\space
  \string"#2\string"%
}
\@onlypremakeg\GlsAddXdyLocation
\else
  \newcommand*\GlsAddXdyLocation[2]{%
    \glsnoinindywarning\GlsAddXdyLocation}
\fi
\ifglsxindy
  \def\@xdylocationclassorder{^^J\space\space\space
    \string"roman-page-numbers\string"^^J\space\space\space
    \string"arabic-page-numbers\string"^^J\space\space\space
    \string"arabic-section-numbers\string"^^J\space\space\space
    \string"alpha-page-numbers\string"^^J\space\space\space
    \string"Roman-page-numbers\string"^^J\space\space\space
    \string"Alpha-page-numbers\string"^^J\space\space\space
    \string"Appendix-page-numbers\string"
    \@xdyuserlocationnames^^J\space\space\space
    \string"see\string"
  }
\fi
\ifglsxindy
  \newcommand*\GlsSetXdyLocationClassOrder[1]{%
    \def\@xdylocationclassorder{#1}}
\else
  \newcommand*\GlsSetXdyLocationClassOrder[1]{%
    \glsnoinindywarning\GlsSetXdyLocationClassOrder}
\fi
\ifglsxindy
  \def\@xdysortrules{}
\fi
\ifglsxindy
  \newcommand*\GlsAddSortRule[2]{%
    \expandafter\toks@\expandafter{\@xdysortrules}%
    \protected@edef\@xdysortrules{\the\toks@ ^^J
      (sort-rule \string"#1\string" \string"#2\string"))}%
  }
\else
  \newcommand*\GlsAddSortRule[2]{%
    \glsnoinindywarning\GlsAddSortRule}

```

```

\fi
\ifglxindy
  \def\@xdyrequiredstyles{tex}
\fi
\ifglxindy
  \newcommand*\GlsAddXdyStyle[1]{%
    \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
  \newcommand*\GlsAddXdyStyle[1]{%
    \glsnoxywarning\GlsAddXdyStyle}
\fi
\ifglxindy
  \newcommand*\GlsSetXdyStyles[1]{%
    \edef\@xdyrequiredstyles{#1}}
\else
  \newcommand*\GlsSetXdyStyles[1]{%
    \glsnoxywarning\GlsSetXdyStyles}
\fi
\def\@xdy@language#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
    \ifglossaryexists{#1}{%
      \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
    }{%
      \PackageError{glossaries}{Can't set language type for
        glossary type '#1' --- no such glossary}{%
        You have specified a glossary type that doesn't exist}}
\else
  \newcommand*\GlsSetXdyLanguage[2][]{%
    \glsnoxywarning\GlsSetXdyLanguage}
\fi
\def\@gls@codepage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyCodePage[1]{%
    \renewcommand*\@gls@codepage{#1}%
  }
  \AtBeginDocument{%
    \ifx\@gls@codepage\@empty
      \ifpackageloaded{fontspec}{\def\@gls@codepage{utf8}}{}%
    \fi
  }
\else
  \newcommand*\GlsSetXdyCodePage[1]{%
    \glsnoxywarning\GlsSetXdyCodePage}
\fi
\ifglxindy
  \ifglx@xindy@glsnumbers
    \def\@xdy@lettergroups{(define-letter-group
      \string"glxnumbers\string"^^J\space\space\space
      :prefixes (\string"0\string" \string"1\string"

```



```

        \string"2\string" \string"3\string" \string"4\string"
        \string"5\string" \string"6\string" \string"7\string"
        \string"8\string" \string"9\string")^^J\space\space\space
        \@xdynumbergrouporder)}
\else
  \def\@xdylettergroups{}
\fi
\fi
\newcommand*\GlsAddLetterGroup[2]{%
  \expandafter\toks@\expandafter{\@xdylettergroups}%
  \protected@edef\@xdylettergroups{\the\toks@^^J%
    (define-letter-group \string"#1\string"^^J\space\space\space#2)}%
  }%
\newcommand*\forallglossaries}[3][\@glo@types]{%
  \@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%
}

\newcommand*\forallacronyms}[2]{%
  \@for#1:=\@glsacronymlists\do{\ifx#1\@empty\else#2\fi}%
}

\newcommand*\forglsentries}[3][\glsdefaulttype]{%
  \protected@edef\@glo@list{\csname glolist@#1\endcsname}%
  \@for#2:=\@glo@list\do
  {%
    \ifdefempty{#2}{-}{#3}%
  }%
}

\newcommand*\forallglsentries}[3][\@glo@types]{%
  \expandafter\forallglossaries\expandafter[#1]{\@@thisglo@}%
  {%
    \forglsentries[\@@thisglo@]{#2}{#3}%
  }%
}

\newcommand{\ifglossaryexists}{%
  \@ifstar\s@ifglossaryexists\@ifglossaryexists
}

\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{@glo@type@#1@out}{#3}{#2}%
}

\newcommand{\s@ifglossaryexists}[3]{%
  \ifcsundef{glolist@#1}{#3}{#2}%
}

\newcommand*\glsdetoklabel}[1]{#1}
\newcommand*\ifglsentryexists}[3]{%
  \ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}

\newcommand*\ifglsused}[3]{%
  \ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}

\newcommand*\glsdoifexists}[2]{%

```

```

\ifglentryexists{#1}{#2}{%
  \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
    has not been defined}{You need to define a glossary entry before you
    can use it.}}%
}
\newcommand{\glsdoifnoexists}[2]{%
  \ifglentryexists{#1}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
      been defined}{}}{#2}%
}
\newcommand{\glsdoifexistsorwarn}[2]{%
  \ifglentryexists{#1}{#2}{%
    \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}%
  }%
}
\newcommand{\glsdoifexistsordo}[3]{%
  \ifglentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
  #3%
}%
}
\newcommand{\doifglossarynoexistsordo}[3]{%
  \ifglossaryexists*{#1}%
  {%
    \PackageError{glossaries}{Glossary type ‘#1’ already exists}{}%
    #3%
  }%
  {#2}%
}
\newrobustcmd{\ifglshaschildren}[3]{%
  \glsdoifexists{#1}%
  {%
    \def\do@glshaschildren{#3}%
    \protected@edef\@gls@thislabel{\glsdetoklabel{#1}}%
    \expandafter\for@glentries\expandafter
      [\csname glo@\@gls@thislabel @type\endcsname]
    {\glo@label}%
    {%
      \letcs@glo@parent{glo@\glo@label @parent}%
      \ifdefequal\@gls@thislabel@glo@parent
        {%
          \def\do@glshaschildren{#2}%
          \@endfortrue
        }%
      }%
    }%
  }%
  \do@glshaschildren
}

```

```

    }%
  }
  \newcommand{\ifglshasparent}[3]{%
    \glsdoifexists{#1}%
    {%
      \ifcsemtyp{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
    }%
  }
  \newcommand*\ifglshasdesc}[3]{%
    \ifcsemtyp{glo@\glsdetoklabel{#1}@desc}%
    {#3}%
    {#2}%
  }
  \newcommand*\ifglshasdescsuppressed}[3]{%
    \ifcsequal{glo@\glsdetoklabel{#1}@desc}{@no@post@desc}%
    {#2}%
    {#3}%
  }
  \newrobustcmd*\ifglshassymbol}[3]{%
    \letcs{\@glo@symbol}{glo@\glsdetoklabel{#1}@symbol}%
    \ifdefempty\@glo@symbol
    {#3}%
    {%
      \ifdefequal\@glo@symbol\@gls@default@value
      {#3}%
      {#2}%
    }%
  }
  \newrobustcmd*\ifglshaslong}[3]{%
    \letcs{\@glo@long}{glo@\glsdetoklabel{#1}@long}%
    \ifdefempty\@glo@long
    {#3}%
    {%
      \ifdefequal\@glo@long\@gls@default@value
      {#3}%
      {#2}%
    }%
  }
  \newrobustcmd*\ifglshasshort}[3]{%
    \letcs{\@glo@short}{glo@\glsdetoklabel{#1}@short}%
    \ifdefempty\@glo@short
    {#3}%
    {%
      \ifdefequal\@glo@short\@gls@default@value
      {#3}%
      {#2}%
    }%
  }
  \newcommand{\ifglsfieldvoid}[4]{\ifcsvoid{glo@\glsdetoklabel{#2}@#1}{#3}{#4}}
  \newrobustcmd*\ifglshasfield}[4]{%

```

```

\glsdoifexists{#2}%
{%
  \letcs{\@glo@thisvalue}{glo@glsetoklabel{#2}@#1}%
  \ifdef\@glo@thisvalue
  {%
    \ifdefempty\@glo@thisvalue
    {%
      #4%
    }%
  }%
  \ifdequal\@glo@thisvalue\@gls@default@value
  {%
    #4%
  }%
  {%
    \let\glscurrentfieldvalue\@glo@thisvalue
    #3%
  }%
}%
}%
{%
  \@gls@fetchfield{\@gls@thisfield}{#1}%
  \ifdef\@gls@thisfield
  {%
    \letcs{\@glo@thisvalue}{glo@glsetoklabel{#2}@ \@gls@thisfield}%
    \ifdefempty\@glo@thisvalue
    {%
      #4%
    }%
    {%
      \ifdequal\@glo@thisvalue\@gls@default@value
      {%
        #4%
      }%
    }%
    \let\glscurrentfieldvalue\@glo@thisvalue
    #3%
  }%
}%
}%
{%
  \GlossariesWarning{Unknown entry field '#1'}%
  #4%
}%
}%
}
\newcommand*{\glscurrentfieldvalue}{ }
\newcommand*{\@glo@types}{,}
\newcommand*{\@gls@provide@newglossary}{%

```

```

\protected@write\@auxout{}\string\providecommand\string\@newglossary[4]{}%
\let\@gls@provide@newglossary\relax
}
\newcommand*{\@gls@defaulttype}{\glsdefaulttype}
\newcommand*{\defglsentryfmt}[2][\glsdefaulttype]{%
\protected@edef\@glo@tmp{#1}%
\ifx\@glo@tmp\@gls@defaulttype
\GlossariesWarning{\string\defglsentryfmt\space attempting to set
a display style for \string\glsdefaulttype\space but there
doesn't seem to be a default glossary. If you have used
'nomain', you need to define an alternative glossary}%
\else
\csgdef{gls@\@glo@tmp @entryfmt}{#2}%
\fi
}
\newcommand*{\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*{\@gls@forbidtext}[1]{%
\ifboolexpr{test {\ifdefstring{#1}{tex}}
or test {\ifdefstring{#1}{TEX}}}
{%
\def#1{nottex}%
\PackageError{glossaries}%
{Forbidden '.tex' extension replaced with '.nottex'}%
{I'm sorry, I can't allow you to do something so reckless.\MessageBreak
Don't use '.tex' as an extension for a temporary file.}%
}%
}%
}%
}
\newcommand*{\gls@gobbleopt}{\new@ifnextchar[{\@gls@gobbleopt}{}]}
\def\@gls@gobbleopt[#1]{
\newcommand*{\newglossary}\@ifstar\s@newglossary\@ns@newglossary}
\newcommand*{\s@newglossary}[2]{%
\@ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
}
\newcommand*{\ns@newglossary}[5][glg]{%
\doifglossarynoexistsordo{#2}%
{%
\ifundef\glsdefaulttype
{%
\gdef\glsdefaulttype{#2}%
}{}%
\toks@{#2}\protected@edef\@glo@types{\@glo@types\the\toks@,}%
\expandafter\gdef\csname glolist@#2\endcsname{,}%
\expandafter\edef\csname @glotype@#2@log\endcsname{#1}%
\expandafter\edef\csname @glotype@#2@in\endcsname{#3}%
\expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
\expandafter\@gls@forbidtext\csname @glotype@#2@log\endcsname
\expandafter\@gls@forbidtext\csname @glotype@#2@in\endcsname
\expandafter\@gls@forbidtext\csname @glotype@#2@out\endcsname

```

```

\expandafter\def\csname @glotype@#2@title\endcsname{#5}%
\@gls@provide@newglossary
\protected@write\@auxout{}\string\@newglossary{#2}{#1}{#3}{#4}}%
\ifcsundef{gls@#2@entryfmt}%
{%
  \defglsentryfmt[#2]{\glsentryfmt}%
}%
{}%
\@gls@defsortcount{#2}%
\@ifnextchar[{\@gls@setcounter{#2}}%
  {\@gls@setcounter{#2}[\glscounter]}%
}%
{%
  \gls@gobbleopt
}%
}
\newcommand*\altnewglossary}[3]{%
  \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
}
\@onlypreamble{\newglossary}
\@onlypremakeg\newglossary
\newcommand*\@newglossary}[4]{%
\def\@gls@setcounter#1[#2]{%
  \expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
  \ifglsxindy
    \GlsAddXdyCounters{#2}%
  \fi
}
\newcommand*\@gls@getcounter}[1]{%
  \csname @glotype@#1@counter\endcsname
}
\glsdefmain
\@gls@do@acronymsdef
\@gls@do@symbolsdef
\@gls@do@numbersdef
\@gls@do@indexdef
\newcommand*\newignoredglossary}[1]{%
  \ifdefempty\@ignored@glossaries
  {%
    \protected@edef\@ignored@glossaries{#1}%
  }%
  {%
    \protected@eappto\@ignored@glossaries{,#1}%
  }%
  \csgdef{glolist@#1}{,}%
  \ifcsundef{gls@#1@entryfmt}%
  {%
    \defglsentryfmt[#1]{\glsentryfmt}%
  }%
  {}%
}

```

```

\ifdefempty\@gls@nohyperlist
{%
  \renewcommand*\@gls@nohyperlist}{#1}%
}%
{%
  \protected@eappto\@gls@nohyperlist{, #1}%
}%
}
\newcommand*\@ignored@glossaries{}
\newcommand*\@ifignoredglossary}[3]{%
  \protected@edef\@gls@igtype{#1}%
  \expandafter\DTLifinlist\expandafter
  {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
}
\define@key{glossentry}{name}{%
\def\@glo@name{#1}%
}
\define@key{glossentry}{description}{%
\def\@glo@desc{#1}%
}
\define@key{glossentry}{descriptionplural}{%
\def\@glo@descplural{#1}%
}
\define@key{glossentry}{sort}{%
\def\@glo@sort{#1}}
\define@key{glossentry}{text}{%
\def\@glo@text{#1}%
}
\define@key{glossentry}{plural}{%
\def\@glo@plural{#1}%
}
\define@key{glossentry}{first}{%
\def\@glo@first{#1}%
}
\define@key{glossentry}{firstplural}{%
\def\@glo@firstplural{#1}%
}
\newcommand*\@gls@default@value{\relax}
\define@key{glossentry}{symbol}{%
\def\@glo@symbol{#1}%
}
\define@key{glossentry}{symbolplural}{%
\def\@glo@symbolplural{#1}%
}
\define@key{glossentry}{type}{%
\def\@glo@type{#1}}
\define@key{glossentry}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%

```

```

    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
  {%
    \def\@glo@counter{#1}%
  }%
}
\define@key{glossentry}{see}{%
  \gls@set@xr@key{see}{\@glo@see}{#1}%
}
\newcommand*\@gls@set@xr@key[3]{%
  \renewcommand*\@gls@xr@key{#1}%
  \gls@checkseeallowed
  \def#2{#3}%
  \@glo@seeautonumberlist
}
\newcommand*\@gls@xr@key{see}
\newcommand*\@gls@checkseeallowed{%
  \@gls@see@noindex
}
\newcommand*\@gls@checkseeallowed@preambleonly{%
  \GlossariesWarning
  {‘\@gls@xr@key’ key doesn’t have any effect when used in the document
  environment. Move the definition to the preamble
  after \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@key{glossentry}{parent}{%
  \def\@glo@parent{#1}}
\define@choicekey{glossentry}{nonumberlist}%
[\@gls@nonumberlist@val\@gls@nonumberlist@nr]{true,false}[true]%
{%
  \ifcase\@gls@nonumberlist@nr\relax
    \def\@glo@prefix{\@glsnonextpages}%
    \@gls@savenonumberlist{true}%
  \else
    \def\@glo@prefix{\@glsnextpages}%
    \@gls@savenonumberlist{false}%
  \fi
}
\newcommand*\@gls@noidx@nonumberlist[1]{%
  \cslet\@glo@\@glo@label @prenumberlist\@glo@prefix
}%
\newcommand{\@glsnoidxprenumberlist}[1]{%
  \csuse{glo@\@glsdetoklabel{#1}@prenumberlist}}
\newcommand*\@gls@savenonumberlist[1]{%
}
\newcommand*\@gls@initnonumberlist{%
}

```



```

\newcommand*{\@gls@storenonumberlist}[1]{%
\newcommand*{\@gls@enablesavenonumberlist}{%
  \renewcommand*{\@gls@initnonumberlist}{%
    \undef\@glo@nonumberlist
  }%
  \renewcommand*{\@gls@savenonumberlist}[1]{%
    \def\@glo@nonumberlist{##1}%
  }%
  \renewcommand*{\@gls@storenonumberlist}[1]{%
    \ifdef\@glo@nonumberlist
    {%
      \cslet{glo@glsdetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
    }%
    {}%
  }%
  \appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}
\define@key{glossentry}{user1}{%
  \def\@glo@useri{#1}%
}
\define@key{glossentry}{user2}{%
  \def\@glo@userii{#1}%
}
\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#1}%
}
\define@key{glossentry}{short}{%
  \def\@glo@short{#1}%
}
\define@key{glossentry}{shortplural}{%
  \def\@glo@shortpl{#1}%
}
\define@key{glossentry}{long}{%
  \def\@glo@long{#1}%
}
\define@key{glossentry}{longplural}{%
  \def\@glo@longpl{#1}%
}
\newcommand*{\@gls@noname}{%
  \PackageError{glossaries}{name key required in
  \string\newglossaryentry\space for entry '\@glo@label'}{You

```

```

haven't specified the entry name}}
\newcommand*\@glsnodesc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '\@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}%
\newcommand*\@gls@missingnumberlist}[1]{%
  ??%
  \ifglssavenumberlist
  \GlossariesWarning{Missing number list for entry '#1'.
    Maybe makeglossaries + rerun required}%
  \else
  \PackageError{glossaries}%
  {Package option 'savenumberlist=true' required}%
  {%
    You must use the 'savenumberlist' package option
    to reference location lists.%
  }%
  \fi
}
\newcommand*\@gls@defaultsort}{\@glo@name}
\newcount\gls@level
\newcommand*\@gls@noexpand@field}[3]{%
  \expandafter\global\expandafter
  \let\csname glo@#1@#2\endcsname#3%
}
\newcommand*\@gls@noexpand@fields}[4]{%
  \ifcsdef{gls@assign@#3@field}
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
    }%
    {%
      \csuse{gls@assign@#3@field}{#2}{#4}%
    }%
  }%
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \@gls@noexpand@field{#2}{#3}{\@gls@value}%
    }%
    {%
  }

```

```

        \@@gls@noexpand@field{#2}{#3}{#4}%
    }%
}
\newcommand{\@@gls@expand@field}[3]{%
\expandafter
\protected@xdef\csname glo@#1@#2\endcsname{#3}%
}
\newcommand{\@gls@expand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
{%
\@@gls@expand@field{#2}{#3}{#4}%
}%
{%
\csuse{gls@assign@#3@field}{#2}{#4}%
}%
}%
}%
\ifdefequal{#4}{\@gls@default@value}%
{%
\@@gls@expand@field{#2}{#3}{#1}%
}%
{%
\@@gls@expand@field{#2}{#3}{#4}%
}%
}%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
\def\@gls@tmp{#1}%
\ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*\{gls@expandfields}{%
\let\gls@assign@field\@gls@expand@fields
}
\newcommand*\{gls@noexpandfields}{%
\let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
\glsdoifnoexists{#1}%

```

```

    {%
      \gls@defglossaryentry{#1}{#2}%
    }%
  }
\newcommand*\gls@defdocnewglossaryentry{%
  \let\newglossaryentry\new@glossaryentry
}
\newrobustcmd{\provideglossaryentry}[2]{%
  \ifglsentryexists{#1}%
  {%
    {%
      \gls@defglossaryentry{#1}{#2}%
    }%
  }
}
\@onlypreamble{\provideglossaryentry}
\newcommand*\gls@glsdefs@labellist{}
\newcommand*\gls@glsdefs@addlabel}[1]{\listxadd{\gls@glsdefs@labellist}{#1}}
\newcommand*\gls@docdef@labellist{}
\newcommand*\gls@docdef@addlabel}[1]{\listxadd{\gls@docdef@labellist}{#1}}
\newrobustcmd{\new@glossaryentry}[2]{%
  \ifundef\@gls@deffile
  {%
    \global\newwrite\@gls@deffile
    \immediate\openout\@gls@deffile=\jobname.glsdefs
  }%
  {%
    \ifglsentryexists{#1}%
    {%
      \xifinlist{#1}{\gls@docdef@labellist}%
      {%
        \GlossariesWarning{Entry ‘#1’ has already been defined in the
          document environment (ignoring new definition)}%
      }%
      {%
        \xifinlist{#1}{\gls@glsdefs@labellist}%
        {%
          \gls@defglossaryentry{#1}{#2}%
          \gls@docdef@addlabel{#1}%
          \@gls@writedef{#1}%
        }%
        {%
          \GlossariesWarning{Entry ‘#1’ has already been defined in
            the preamble (in which case remove the duplicate definition)
            or you have just updated to glossaries.sty
            v4.47 (in which case delete ‘\jobname.glsdefs’)}%
        }%
      }%
    }%
  }%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}

```

```

        \gls@docdef@addlabel{#1}%
        \@gls@writedef{#1}%
    }%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*\gls@begindocdefs{%
    \gls@enablesavenonumberlist
    \edef\@gls@restreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
    \makeatletter
    \InputIfFileExists{\jobname.glsdefs}{-}{-}%
    \@gls@restreat
    \undef\@gls@restreat
    \gls@defdocnewglossaryentry
}
\newcommand*\glsdefs@newdocentry}[2]{%
    \ifglsentryexists{#1}%
    {\GlossariesWarning{entry ‘#1’ has already been defined in the preamble
        (if you have moved the definition there from the document
        environment, delete \jobname.glsdefs)}}%
    {%
        \gls@glsdefs@addlabel{#1}%
        \gls@defglossaryentry{#1}{#2}%
    }%
}
}
\newcommand*\@gls@writedef}[1]{%
    \immediate\write\@gls@deffile
    {%
        \string\glsdefs@newdocentry{\glsdetoklabel{#1}}\glspercentchar^^J%
        \expandafter\@gobble\string\{\glspercentchar%
    }%
    \@for\@gls@map:=\@gls@keymap\do
    {%
        \letcs\glo@value{glo@\glsdetoklabel{#1}}\expandafter\@secondoftwo\@gls@map}%
        \ifdef\glo@value
        {%
            \@onelevel@sanitize\glo@value
            \immediate\write\@gls@deffile
            {%
                \expandafter\@firstoftwo\@gls@map
                =\expandafter\@gobble\string\{\glo@value\expandafter\@gobble\string\},%
                \glspercentchar
            }%
        }%
    }%
    }%
}
\gls@writedefhook
\immediate\write\@gls@deffile
{%
    \glspercentchar^^J%
}

```

```

    \expandafter\@gobble\string\}\glspercentchar^^J%
  }%
}
\newcommand*{\@gls@keymap}{%
  {name}{name},%
  {sort}{sortvalue},% unescaped sort value
  {type}{type},%
  {first}{first},%
  {firstplural}{firstpl},%
  {text}{text},%
  {plural}{plural},%
  {description}{desc},%
  {descriptionplural}{descplural},%
  {symbol}{symbol},%
  {symbolplural}{symbolplural},%
  {user1}{useri},%
  {user2}{userii},%
  {user3}{useriii},%
  {user4}{useriv},%
  {user5}{userv},%
  {user6}{uservi},%
  {long}{long},%
  {longplural}{longpl},%
  {short}{short},%
  {shortplural}{shortpl},%
  {counter}{counter},%
  {parent}{parent}%
}
\newcommand*{\@gls@fetchfield}[2]{%
  \edef\@gls@thisval{#2}%
  \@for\@gls@map:=\@gls@keymap\do{%
    \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
    \ifdefequal{\@this@key}{\@gls@thisval}%
    {%
      \edef#1{\expandafter\@secondoftwo\@gls@map}%
      \@endfortrue
    }%
  }%
}%
}
\newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
\newcommand*{\@sglsaddstoragekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
}%
}

```

```

\@glsaddstoragekey{#1}%
}
\newcommand*{\@glsaddstoragekey}[3]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{\@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
}%
{%
\PackageError{glossaries}{Key ‘#1’ already exists}{}%
}%
}
\newcommand*{\glsaddkey}{\@ifstar\@sglsaddkey\@glsaddkey}
\newcommand*{\@sglsaddkey}[1]{%
\key@ifundefined{glossentry}{#1}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname gls@assign@#1@field\endcsname}[2]{%
\@gls@expand@field{##1}{#1}{##2}%
}%
}%
}%
\@glsaddkey{#1}%
}
\newcommand*{\@glsaddkey}[7]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{\@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
\newcommand*{#4}[1]{\@Glsentryfield{##1}{#1}}%
\ifcsdef{@gls@user@#1@}%
{%
\PackageError{glossaries}%
{Can’t define ‘\string#5’ as helper command
‘\expandafter\string\csname @gls@user@#1@\endcsname’ already exists}%
}%
}%
}%

```

```

\expandafter\newcommand\expandafter*\expandafter
  {\csname @gls@user@#1\endcsname}[2][]{%
    \new@ifnextchar[%
      {\csuse{@gls@user@#1@}{##1}{##2}}%
      {\csuse{@gls@user@#1@}{##1}{##2}[]}}%
\csdef{@gls@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{#3{##2}##3}%
}%
\newrobustcmd*{#5}{%
  \expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
}%
\ifcsdef{@Gls@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#6' as helper command
  '\expandafter\string\csname @Gls@user@#1\endcsname' already exists}%
  {}%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
  {\csname @Gls@user@#1\endcsname}[2][]{%
    \new@ifnextchar[%
      {\csuse{@Gls@user@#1@}{##1}{##2}}%
      {\csuse{@Gls@user@#1@}{##1}{##2}[]}}%
\csdef{@Gls@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{#4{##2}##3}%
}%
\newrobustcmd*{#6}{%
  \expandafter\@gls@hyp@opt\csname @Gls@user@#1\endcsname}%
}%
\ifcsdef{@GLS@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#7' as helper command
  '\expandafter\string\csname @GLS@user@#1\endcsname' already exists}%
  {}%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
  {\csname @GLS@user@#1\endcsname}[2][]{%
    \new@ifnextchar[%
      {\csuse{@GLS@user@#1@}{##1}{##2}}%
      {\csuse{@GLS@user@#1@}{##1}{##2}[]}}%
\csdef{@GLS@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{\glsuppercase{#3{##2}##3}}%
}%
\newrobustcmd*{#7}{%
  \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
}%
\glsmfuaddmap{#3}{#4}%

```



```

    \glsmfuaddmap{#5}{#6}%
    \glsmfublocker{#7}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#1’ already exists}{}%
  }%
}
\newrobustcmd{\glsfieldxdef}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \protected@csxdef{glo@\@glo@label @#2}{#3}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newrobustcmd{\glsfieldedef}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \protected@csedef{glo@\@glo@label @#2}{#3}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newrobustcmd{\glsfieldgdef}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newrobustcmd{\glsfieldddef}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%

```

```

\ifcsdef{glo@\@glo@label @#2}%
{%
  \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
}%
{%
  \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
}%
}%
}
\newrobustcmd{\glsfieldfetch}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \letcs#3{glo@\@glo@label @#2}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand*\glsunexpandedfieldvalue[2]{%
  \ifcsname glo@\glsdetoklabel{#1}@#2\endcsname
  \expandafter\expandonce\csname glo@\glsdetoklabel{#1}@#2\expandafter\endcsname
  \fi
}
\newrobustcmd{\ifglsfieldeq}[5]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newrobustcmd{\ifglsfielddefeq}[5]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \expandafter\ifdefstrequal\csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}

```

```

    }%
  }%
}
\newcommand{\ifglsfieldcseq}[5]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\glo@label @#2}%
    {%
      \ifcsstrequal{glo@\glo@label @#2}{#3}{#4}{#5}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand*{\gls.writedefhook}{}
\newcommand*{\gls@assign@desc}[1]{%
  \gls@assign@field{#1}{desc}{\@glo@desc}%
  \gls@assign@field{\@glo@desc}{#1}{descplural}{\@glo@descplural}%
}
\newcommand{\longnewglossaryentry}[3]{%
  \glsdoifnoexists{#1}%
  {%
    \bgroup
    \let\@org@newglossaryentryprehook\@newglossaryentryprehook
    \long\def\@newglossaryentryprehook{%
      \long\def\@glo@desc{#3\leavevmode\unskip\nopostdesc}%
      \@org@newglossaryentryprehook
    }%
    \renewcommand*{\gls@assign@desc}[1]{%
      \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
      \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@desc}%
    }
    \gls@defglossaryentry{#1}{#2}%
  \egroup
}
}
\@onlypreamble{\longnewglossaryentry}
\newcommand{\longprovideglossaryentry}[3]{%
  \ifglsentryexists{#1}{%
    {\longnewglossaryentry{#1}{#2}{#3}}%
  }
}
\@onlypreamble{\longprovideglossaryentry}
\newcommand{\gls@defglossaryentry}[2]{%
  \let\GlsSetQuote\gls@nosetquote
  \protected@edef\glo@label{\glsdetoklabel{#1}}%
  \let\glslabel\@glo@label
  \let\@glo@name\@glsname
  \let\@glo@desc\@glsnodesc

```

```

\let\@glo@descplural\@gls@default@value
\let\@glo@type\@gls@default@value
\let\@glo@symbol\@gls@default@value
\let\@glo@symbolplural\@gls@default@value
\let\@glo@text\@gls@default@value
\let\@glo@plural\@gls@default@value
\let\@glo@first\@gls@default@value
\let\@glo@firstplural\@gls@default@value
\let\@glo@sort\@gls@default@value
\let\@glo@counter\@gls@default@value
\def\@glo@see{ }%
\def\@glo@parent{ }%
\def\@glo@prefix{ }%
\@gls@initnonumberlist
\def\@glo@useri{ }%
\def\@glo@userii{ }%
\def\@glo@useriii{ }%
\def\@glo@useriv{ }%
\def\@glo@userv{ }%
\def\@glo@uservi{ }%
\def\@glo@short{ }%
\def\@glo@shortpl{ }%
\def\@glo@long{ }%
\def\@glo@longpl{ }%
\@newglossaryentryprehook
\setkeys{glossentry}{#2}%
\ifundef\glsdefaulttype
{%
  \PackageError{glossaries}%
  {No default glossary type (have you used ‘nomain’ by mistake?)}%
  {If you use package option ‘nomain’ you must define
   a new glossary before you can define entries}%
}%
}%
\gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
\protected@edef\@glo@type{\glsentrytype{\@glo@label}}%
\ifcsundef{glo@list@\@glo@type}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘\@glo@type’ has not been defined}%
  {You need to define a new glossary type, before making entries
   in it}%
}%
\ifignoredglossary\@glo@type
{%
  \ifx\@glo@desc\@glsnodesc
    \let\@glo@desc\@empty
  \fi
}%

```

```

    {%
    }%
    \protected@edef\@glo@list@\csname glo@list@\@glo@type\endcsname}%
    \expandafter\protected@xdef\csname glo@list@\@glo@type\endcsname{%
      \@glo@list@\@glo@label},}%
  }%
  \gls@level=0\relax
  \ifx\@glo@parent\@empty
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
  \else
    \ifdefequal\@glo@label\@glo@parent%
      {%
      \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
      \def\@glo@parent{}%
      \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
      }%
      {%
      \ifglsentryexists{\@glo@parent}%
      {%
      \expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
        \@glo@parent}%
      \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
      \advance\gls@level by 1\relax
      \ifx\@glo@name\@gls@name
        \expandafter\let\expandafter\@glo@name
          \csname glo@\@glo@parent @name\endcsname
      \ifx\@glo@plural\@gls@default@value
        \expandafter\let\expandafter\@glo@plural
          \csname glo@\@glo@parent @plural\endcsname
      \fi
      \fi
      }%
      {%
      \PackageError{glossaries}%
      {%
      Invalid parent ‘\@glo@parent’
      for entry ‘\@glo@label’ - parent doesn’t exist%
      }%
      {%
      Parent entries must be defined before their children%
      }%
      \def\@glo@parent{}%
      \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
      }%
      }%
    \fi
    \expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%
    \gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
    \letcs\@glo@sort{glo@\@glo@label @sortvalue}%
    \gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%
  
```

```

\expandafter\gls@assign@field\expandafter
  {\csname glo@\glo@label @text\endcsname\glspluralsuffix}%
  {\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\glo@label @text\endcsname}%
  {\@glo@label}{first}{\@glo@first}%
\ifx\@glo@first\@gls@default@value
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\glo@label @plural\endcsname}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\else
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\glo@label @first\endcsname\glspluralsuffix}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\fi
\ifcsundef{@glo@type@\@glo@type @counter}%
{%
  \def\@glo@defaultcounter{\glscounter}%
}%
{%
  \letcs\@glo@defaultcounter{@glo@type@\@glo@type @counter}%
}%
\gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
\gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
\gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
\gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
\gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
\gls@assign@field{}{\@glo@label}{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\@gls@noname
  \@gls@noname
  \let\@glo@name\@gls@default@value
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%
\ifcsundef{glo@\glo@label @numberlist}%
{%
  \csxdef{glo@\glo@label @numberlist}{%
    \noexpand\@gls@missingnumberlist{\@glo@label}}%
}%
{%
\@gls@storenonumberlist{\@glo@label}%
\def\@glo@@@desc{\@glo@first}%
\ifx\@glo@desc\@glo@@@desc
  \let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@gls@nodesc

```

```

    \@glsnodesc
    \let\@glsdesc\@gls@default@value
\fi
\gls@assign@desc{\@glo@label}%
\@gls@defsort{\@glo@type}{\@glo@label}%
\def\@glo@@symbol{\@glo@text}%
\ifx\@glo@symbol\@glo@@symbol
    \let\@glo@symbol\@glo@text
\fi
\gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
\expandafter
    \gls@assign@field\expandafter
    {csname glo@\@glo@label @symbol\endcsname}
    {\@glo@label}{symbolplural}{\@glo@symbolplural}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagfalse\endcsname{%
\noexpand\global
    \noexpand\let\expandafter\noexpand
        \csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagtrue\endcsname{%
\noexpand\global
    \noexpand\let\expandafter\noexpand
        \csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
}%
\csname glo@\@glo@label @flagfalse\endcsname
\@glo@autosee
\ifignoredglossary\@glo@type
{%
    \csdef{glo@\@glo@label @index}{}%
}
{%
    \do@glo@storeentry{\@glo@label}%
}%
\@newglossaryentry@defcounters
\@newglossaryentryposthook
}
\newcommand*{\@glo@autosee}{%
    \ifdefvoid\@glo@see{%
        {%
            \protected@edef\@do@glssee{%
                \noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
                \noexpand\expandafter\noexpand\@glssee\noexpand\@glo@list{\@glo@label}}%
            \@do@glssee
        }%
        \@glo@autoseehook
    }%
}
\newcommand*{\@glo@autoseehook}{%
}
\newcommand*{\@newglossaryentryprehook}{%
}
\newcommand*{\@newglossaryentryposthook}{%
}
\newcommand*{\@newglossaryentrydefcounters}{%
}

```

```

\newcommand*\glsmoveentry}[2]{%
  \protected@edef\glo@thislabel{\glsdetoklabel{#1}}%
  \protected@edef\glo@type{\csname glo@\glo@thislabel @type\endcsname}%
  \def\glo@list{,%}
  \forlslentries[\glo@type]{\glo@label}%
  {%
    \ifdefequal\glo@thislabel\glo@label
      {\protected@eappto\glo@list{\glo@label,}}%
    }%
  \cslet\glo@list@\glo@type{\glo@list}%
  \csdef\glo@\glo@thislabel @type}{#2}%
}
\ifglxindy
  \newcommand*\@glossaryentryfield{\string\glossentry}
\else
  \newcommand*\@glossaryentryfield{\string\glossentry}
\fi
\ifglxindy
  \newcommand*\@glossarysubentryfield{%
    \string\subglossentry}
\else
  \newcommand*\@glossarysubentryfield{%
    \string\subglossentry}
\fi
\newcommand{\glo@storeentry}[1]{%
  \protected@edef\glo@esclabel{#1}%
  \gls@checkmkidxchars\glo@esclabel
  \protected@edef\glo@sort{\csname glo@#1@sort\endcsname}%
  \gls@checkmkidxchars\glo@sort
  \gls@checkmkidxchars\glo@prefix
  \protected@edef\glo@parent{\csname glo@#1@parent\endcsname}%
  \ifglxindy
    \ifx\glo@parent\@empty
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        (\string"\glo@sort\string" %
        \string"\glo@prefix\@glossaryentryfield{\glo@esclabel}\string") %
      }%
    \else
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \csname glo@\glo@parent @index\endcsname
        (\string"\glo@sort\string" %
        \string"\glo@prefix\@glossarysubentryfield
        {\csname glo@#1@level\endcsname}{\glo@esclabel}\string") %
      }%
    \fi
  \else
    \ifx\glo@parent\@empty
      \@onelevel@sanitize\glo@prefix
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \glo@sort\gls@actualchar\glo@prefix

```



```

        \@glossaryentryfield{\@glo@esclabel}%
    }%
\else
\expandafter\protected\xdef\csname glo@#1@index\endcsname{%
\csname glo@\@glo@parent @index\endcsname\@gls@levelchar
\@glo@sort\@gls@actualchar\@glo@prefix
\@glossarysubentryfield
{\csname glo@#1@level\endcsname}\@glo@esclabel}%
}%
\fi
\fi
}
\newcommand*\@gls@if@measuring}[2]{%
\ifmeasuring@
#1%
\else
#2%
\fi
}
\let\gls@org@refstepcounter\refstepcounter
\let\gls@org@label\label
\newcommand*\gls@if@measuring}[2]{#2}
\AtBeginDocument{%
\@ifpackageloaded{amsmath}%
{\let\gls@if@measuring\@gls@if@measuring}%
{}%
\let\gls@org@refstepcounter\refstepcounter
\let\gls@org@label\label
}
\newcommand*\gls@ifnotmeasuring[1]{\glsifmeasuring}{#1}}
\newcommand*\glsifmeasuring{\gls@if@measuring}
\newcommand*\gls@local@stepcounter}[1]{%
\ifcsdef{c@#1}%
{%
\epreto\gls@measuring@reset{\noexpand\setcounter{#1}{\number\value{#1}}}%
\expandafter\advance\csname c@#1\endcsname by \@ne
}%
{}%
}
\newcommand*\gls@start@measuring}{%
\ifx\glsifmeasuring\@firstoftwo
\else
\let\gls@org@target\glstarget
\let\gls@org@link\glslink
\def\gls@measuring@reset{}%
\fi
\let\glsifmeasuring\@firstoftwo
\let\refstepcounter\gls@local@stepcounter
\let\label\@gobble
\let\glstarget\@secondoftwo

```

```

\let\@glslink\@secondoftwo
}
\newcommand{\gls@stop@measuring}{%
\renewcommand*\gls@ifmeasuring{\gls@if@measuring}%
\let\refstepcounter\gls@org@refstepcounter
\let\label\gls@org@label
\let\glstarget\gls@org@target
\let\@glslink\gls@org@link
\gls@measuring@reset
\def\gls@measuring@reset{%
}
\newcommand{\glsmeasureheight}[2]{%
\settoheight{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\newcommand{\glsmeasuredepth}[2]{%
\settodepth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\newcommand{\glsmeasurewidth}[2]{%
\settowidth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\def\@gls@patchtabularx#1\hbox#2#3!!{%
\def\TX@trial##1{#1\hbox{\gls@start@measuring#2\gls@stop@measuring}#3}%
}
\newcommand*\glspatchtabularx{%
\ifdef\TX@trial
{%
\expandafter\@gls@patchtabularx\TX@trial{##1}!!%
\let\glspatchtabularx\relax
}%
}%
}
\newcommand*\glsreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glsreset{#1}%
}%
}%
}
\newcommand*\glslocalreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glslocalreset{#1}%
}%
}%
}
\newcommand*\glsunset}[1]{%

```

```

\gls@ifnotmeasuring
{%
  \glsdoifexists{#1}%
  {%
    \@glsunset{#1}%
  }%
}%
}
\newcommand*{\glslocalunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glslocalunset{#1}%
    }%
  }%
}
\newcommand*{\@glslocalunset}{\@glslocalunset}
\newcommand*{\@glslocalunset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
}
\newcommand*{\@glsunset}{\@glsunset}
\newcommand*{\@glsunset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
}
\newcommand*{\@glslocalreset}{\@glslocalreset}
\newcommand*{\@glslocalreset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
}
\newcommand*{\@glsreset}{\@glsreset}
\newcommand*{\@glsreset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
}
\newcommand*{\glsresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsreset{\@glsentry}%
  }%
}
\newcommand*{\glslocalresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalreset{\@glsentry}%
  }%
}
\newcommand*{\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsunset{\@glsentry}%
  }%
}

```

```

}
\newcommand*\glslocalunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalunset{\@glsentry}%
  }%
}
\newcommand*\@newglossaryentry@defcounters{%
  \csdef{glo@\@glo@label @currcount}{0}%
  \csdef{glo@\@glo@label @prevcount}{0}%
}
\newif\ifglsresetcurrcount
\glsresetcurrcountfalse
\newcommand*\glsenableentrycount{%
  \let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
  \renewcommand*\gls@defdocnewglossaryentry{%
    \renewcommand*\newglossaryentry[2]{%
      \PackageError{glossaries}{\string\newglossaryentry\space
        may only be used in the preamble when entry counting has
        been activated}{If you use \string\glsenableentrycount\space
        you must place all entry definitions in the preamble not in
        the document environment}%
    }%
  }%
}
\newcommand*\glsentrycurrcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
  {0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*\glsentryprevcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
  {0}{\@gls@entry@field{##1}{prevcount}}%
}%
\renewcommand*\@glsunset}[1]{%
  \@glsunset{##1}%
  \@gls@increment@currcount{##1}%
}%
\renewcommand*\@glslocalunset}[1]{%
  \@glslocalunset{##1}%
  \@gls@local@increment@currcount{##1}%
}%
\renewcommand*\@glsreset}[1]{%
  \@glsreset{##1}%
  \ifglsresetcurrcount\csgdef{glo@\glsdetoklabel{##1}@currcount}{0}\fi
}%
\renewcommand*\@glslocalreset}[1]{%
  \@glslocalreset{##1}%
  \ifglsresetcurrcount\csdef{glo@\glsdetoklabel{##1}@currcount}{0}\fi
}%
\def\@cgls###1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax

```

```

        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \@gls@{##1}{##2}[##3]%
    \fi
}%
\def\@cgl@##1##2[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \@cgl@{##1}{##2}[##3]%
    \fi
}%
\def\@Gls@##1##2[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \@Gls@{##1}{##2}[##3]%
    \fi
}%
\def\@Glspl@##1##2[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \@Glspl@{##1}{##2}[##3]%
    \fi
}%
\AtEndDocument{\@gls@write@entrycounts}%
\renewcommand*{\@gls@entry@count}[2]{%
    \csdef{glo@glsdetoklabel{##1}@prevcount}{##2}%
}%
\let\glsetenableentrycount\relax
}
\@onlypreamble\glsetenableentrycount
\newcommand*{\@gls@increment@currcount}[1]{%
    \csxdef{glo@glsdetoklabel{##1}@currcount}{%
        \number\numexpr\glsetcurrcount{##1}+1}%
}
\newcommand*{\@gls@local@increment@currcount}[1]{%
    \csedef{glo@glsdetoklabel{##1}@currcount}{%
        \number\numexpr\glsetcurrcount{##1}+1}%
}
\newcommand*{\@gls@write@entrycounts}{%
    \immediate\write\@auxout
        {\string\providecommand*{\string\@gls@entry@count}[2]{}}%
    \forallglsentries{\@glsentry}{%
        \ifglsused{\@glsentry}%

```

```

        {\immediate\write\@auxout
         {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}%
        }%
    }%
}
\newcommand*{\@gls@entry@count}[2]{%
\newrobustcmd*{\cgl}{\@gls@hyp@opt\@cgl}
\newcommand*{\@cgl}[2][{}]{%
    \new@ifnextchar[{\@cgl@{#1}{#2}}{\@cgl@{#1}{#2}[]}%
}
\def\@cgl@#1#2[#3]{%
    \GlossariesWarning{\string\cgl\space is defaulting to
    \string\gl\space since you haven't enabled entry counting}%
    \@cgl@{#1}{#2}[#3]%
}
\newcommand*{\cglformat}[2]{%
    \ifglshaslong{#1}{\glsentrylong{#1}}{\glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cGls}{\@gls@hyp@opt\@cGls}
\glsmfuaddmap{\cgl}{\cGls}
\newcommand*{\@cGls}[2][{}]{%
    \new@ifnextchar[{\@cGls@{#1}{#2}}{\@cGls@{#1}{#2}[]}%
}
\def\@cGls@#1#2[#3]{%
    \GlossariesWarning{\string\cGls\space is defaulting to
    \string\Gls\space since you haven't enabled entry counting}%
    \@Gls@{#1}{#2}[#3]%
}
\newcommand*{\cGlsformat}[2]{%
    \ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cglspl}{\@gls@hyp@opt\@cglspl}
\newcommand*{\@cglspl}[2][{}]{%
    \new@ifnextchar[{\@cglspl@{#1}{#2}}{\@cglspl@{#1}{#2}[]}%
}
\def\@cglspl@#1#2[#3]{%
    \GlossariesWarning{\string\cglspl\space is defaulting to
    \string\glspl\space since you haven't enabled entry counting}%
    \@glspl@{#1}{#2}[#3]%
}
\newcommand*{\cglsplformat}[2]{%
    \ifglshaslong{#1}{\glsentrylongpl{#1}}{\glsentryfirstplural{#1}}#2%
}
\newrobustcmd*{\cGlspl}{\@gls@hyp@opt\@cGlspl}
\glsmfuaddmap{\cglspl}{\cGlspl}
\newcommand*{\@cGlspl}[2][{}]{%
    \new@ifnextchar[{\@cGlspl@{#1}{#2}}{\@cGlspl@{#1}{#2}[]}%
}
\def\@cGlspl@#1#2[#3]{%
    \GlossariesWarning{\string\cGlspl\space is defaulting to

```

```

\string\Glspl\space since you haven't enabled entry counting}%
\@Glspl@{#1}{#2}[#3]%
}
\newcommand*\cGlsplformat}[2]{%
\ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
}
\newcommand*\loadglsentries}[2][\@gls@default]{%
\let\@gls@default\glsdefaulttype
\def\glsdefaulttype{#1}\input{#2}%
\let\glsdefaulttype\@gls@default
}
\@onlypreamble{\loadglsentries}
\newcommand*\glstextformat}[1]{#1}
\newcommand*\glsentryfmt{\glsgenentryfmt}
\newcommand*\glsgenentryfmt{%
\ifdefempty\glscustomtext
{%
\glsifplural
{%
\glscapscase
{%
\ifglsused\glslabel
{%
\glsentryplural{\glslabel}\glsinsert
}%
{%
\glsentryfirstplural{\glslabel}\glsinsert
}%
}%
{%
\ifglsused\glslabel
{%
\Glsentryplural{\glslabel}\glsinsert
}%
{%
\Glsentryfirstplural{\glslabel}\glsinsert
}%
}%
{%
\ifglsused\glslabel
{%
\glsuppercase{\glsentryplural{\glslabel}\glsinsert}%
}%
{%
\glsuppercase{\glsentryfirstplural{\glslabel}\glsinsert}%
}%
}%
}%
{%
\glscapscase

```

```

    {%
      \ifglused\glslabel
      {%
        \glsentrytext{\glslabel}\glsinsert
      }%
      {%
        \glsentryfirst{\glslabel}\glsinsert
      }%
    }%
    {%
      \ifglused\glslabel
      {%
        \Glsentrytext{\glslabel}\glsinsert
      }%
      {%
        \Glsentryfirst{\glslabel}\glsinsert
      }%
    }%
    {%
      \ifglused\glslabel
      {%
        \glsuppercase{\glsentrytext{\glslabel}\glsinsert}%
      }%
      {%
        \glsuppercase{\glsentryfirst{\glslabel}\glsinsert}%
      }%
    }%
  }%
}
\newcommand*{\glsacronymfont}{%
  \ifempty\glsacronymfont
  {%
    \ifglused\glslabel
    {%
      \glsifplural
      {%
        \glsacronymfont{\glsentryshortpl{\glslabel}}\glsinsert
      }%
      {%
        \glsacronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
      }%
    }%
  }%
}

```



```

\newcommand*\genacrfullformat}[2]{%
  \glsentrylong{#1}#2\space
  (\protect\firstacronymfont{\glsentryshort{#1}})%
}
\newcommand*\Genacrfullformat}[2]{%
  \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}
\newcommand*\genplacrfullformat}[2]{%
  \glsentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glsentryshortpl{#1}})%
}
\newcommand*\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}
\define@key{glslink}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
  {%
    \def\@gls@counter{#1}%
  }%
}
\define@key{glslink}{format}{%
  \def\@glsnumberformat{#1}}
\newcommand*\@gls@glslink@hyper@update@hook{}
\define@boolkey{glslink}{hyper}[true]{\@gls@glslink@hyper@update@hook}
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\newcommand*\@gls@save@glslocal{%
  \let\if@org@KV@glslink@local\ifKV@glslink@local
}
\newcommand*\@gls@restore@glslocal{%
  \ifKV@glslink@local
    \let\@gls@do@glsunset@glslocalunset
  \else
    \let\@gls@do@glsunset@glsunset
  \fi
  \let\ifKV@glslink@local\if@org@KV@glslink@local
}
\newcommand*\@gls@do@glsunset}[1]{\glsunset{#1}}
\define@boolkey{glslink}{local}[true]{}
\@gls@save@glslocal
\newcommand*\glslinkvar}[3]{#1}

```

```

\newcommand*{\@gls@hyp@opt}[1]{%
  \let\glslinkvar\@firstofthree
  \let\@gls@hyp@opt@cs#1\relax
  \@ifstar{\s@gls@hyp@opt}%
  {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*{\s@gls@hyp@opt}[1] []{%
  \let\glslinkvar\@secondofthree
  \@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1] []{%
  \let\glslinkvar\@thirdofthree
  \@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
  \@gls@hyp@opt\@gls@@link
}
\newcommand*{\@gls@@link}[3] []{%
  \glsdoifexistsordo{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \@gls@link[#1]{#2}{#3}%
  }{%
    \glstextformat{#3}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Glslink}{%
  \@gls@hyp@opt\@Gls@@link
}
\newcommand*{\@Gls@@link}[3] []{%
  \@gls@@link[#1]{#2}{\glsentencecase{#3}}%
}
\glsmfuaddmap{\glslink}{\Glslink}
\newcommand*{\glspostlinkhook}{}
\newcommand*{\@gls@link@checkfirsthyper}{%
  \ifglsused{\glslabel}%
  {%
  }%
  {%
    \gls@checkisacronymlist\glstype
    \ifglshyperfirst
      \if@glsisacronymlist
        \ifglsacrfootnote
          \KV@glslink@hyperfalse
        \fi
      \fi
    \else
      \KV@glslink@hyperfalse
    \fi
  }%
  \glslinkcheckfirsthyperhook
}

```

```

}
\newcommand*\glslinkcheckfirsthyperhook{}
\newcommand*\glslinkpostsetkeys{}
\newcommand{\glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}
\newcommand*\do@glsdisablehyperinlist{%
  \expandafter\DTLifinlist\expandafter{\gls\type}{\@gls@nohyperlist}%
  {\KV@glslink@hyperfalse}{}}%
}
\newcommand*\@gls@setdefault@glslink@opts{}
\def\@gls@link[#1]#2#3{%
  \leavevmode
  \protected@edef\glslabel{\glsdetoklabel{#2}}%
  \def\@gls@link@opts{#1}%
  \let\@gls@link@label\glslabel
  \def\@glsnumberformat{\glsnumberformat}%
  \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  \protected@edef\gls\type{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
  \@gls@save@glslocal
  \@gls@setdefault@glslink@opts
  \do@glsdisablehyperinlist
  \do@gls@link@checkfirsthyper
  \setkeys{glslink}{#1}%
  \glslinkpostsetkeys
  \@gls@saveentrycounter
  \@gls@setsort{\glslabel}%
  \glsencapwrcontent{\do@wrglossary{#2}}%
  \ifKV@glslink@hyper
    \@glslink{\glolinkprefix\glslabel}{\gls\textformat{#3}}%
  \else
    \glsdonohyperlink{\glolinkprefix\glslabel}{\gls\textformat{#3}}%
  \fi
  \let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
  \@gls@restore@glslocal
}
\newcommand*\glsencapwrcontent}[1]{#1}
\newcommand*\glolinkprefix{glo:}
\def\glsentrycounter{\glscounter}%
\newcommand*\@gls@saveentrycounter{%
  \def\@gls@Hcounter{}%
  \ifthenelse{\equal{\@gls@counter}{equation}}%
  {
    \ifcsundef{xatlevel@}%
    {%
      \edef\theglsentrycounter{\expandafter\noexpand
        \csname the\@gls@counter\endcsname}%
    }%
  }%
  \ifx\xatlevel@\@empty
    \edef\theglsentrycounter{\expandafter\noexpand

```

```

        \csname the\@gls@counter\endcsname}%
\else
  \savecounters@
  \advance\c@equation by 1\relax
  \edef\theglentrycounter{\csname the\@gls@counter\endcsname}%
  \ifcsundef{theH\@gls@counter}%
  {%
    \def\@gls@Hcounter{\theglentrycounter}%
  }%
  {%
    \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
  }%
  \protected@edef\theHglentrycounter{\@gls@Hcounter}%
  \restorecounters@
\fi
}%
}%
{%
  \edef\theglentrycounter{\expandafter\noexpand
    \csname the\@gls@counter\endcsname}%
}%
\ifx\@gls@Hcounter\@empty
  \ifcsundef{theH\@gls@counter}%
  {%
    \def\theHglentrycounter{\theglentrycounter}%
  }%
  {%
    \protected@edef\theHglentrycounter{\expandafter\noexpand
      \csname theH\@gls@counter\endcsname}%
  }%
\fi
}
\def\@set@glo@numformat#1#2#3#4{%
  \expandafter\@glo@check@mkidxrangear#3\@nil
  \protected@edef#1{%
    \@glo@prefix setentrycounter[#4]{#2}%
    \expandafter\string\csname\@glo@suffix\endcsname
  }%
  \@gls@checkmkidxchars#1%
}
\def\@glo@check@mkidxrangear#1#2\@nil{%
\if#1(\relax
  \def\@glo@prefix{()}%
  \if\relax#2\relax
    \def\@glo@suffix{glsnumberformat}%
  \else
    \def\@glo@suffix{#2}%
  \fi
\else
  \if#1)\relax

```

```

\def\@glo@prefix{)}%
\if\relax#2\relax
\def\@glo@suffi{x{glsnumberformat}}%
\else
\def\@glo@suffi{x{#2}}%
\fi
\else
\def\@glo@prefix{ }\def\@glo@suffi{x{#1#2}}%
\fi
\fi}
\newcommand*{\@gls@escbsdq}[1]{%
\def\@gls@checkedmkidx{}%
\let\gls@xdystring=#1\relax
\@onelevel@sanitize\gls@xdystring
\edef\do@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\@null}%
\do@gls@xdycheckbackslash
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\@null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\ifgls@esclocations
\@for\@gls@tmp:=\gls@protected@pagefmts\do
{%
\edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\ \expandonce\@gls@tmp}%
\@onelevel@sanitize\@gls@sanitized@tmp
\edef\gls@dostsubst{%
\noexpand\DTLsubstituteall\noexpand\gls@xdystring
{\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
}%
\gls@dostsubst
}%
\fi
\let#1=\gls@xdystring
}
\newcommand{\@gls@checkmkidxchars}[1]{%
\ifgls@xindy
\@gls@escbsdq{#1}%
\else
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkquote#1\@nil""\@null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkescquote#1\@nil""\@null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkescactual#1\@nil\?\?\@null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%

```

```

\expandafter\@gls@checkactual#1\@nil??\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkbar#1\@nil||\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkesbar#1\@nil|||\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checklevel#1\@nil!!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkesclevel#1\@nil!!\!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\edef\@do@gls@checkesc{%
  \noexpand\@gls@checkesc\expandonce{#1}\noexpand\@nil
  \@backslashchar\@backslashchar\noexpand\null
}%
\do@gls@checkesc
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\fi
}
\def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
\newtoks\@gls@tmpb
\def\@gls@checkquote#1"#2"#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
  \ifx\null#3\null
  \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
  \def\@@gls@checkquote{\relax}%
  \else
  \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
    \@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
  \def\@@gls@checkquote{\@gls@checkquote#3\null}%
  \fi
  \else
  \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
    \@gls@quotechar\@gls@quotechar}%
  \ifx\null#3\null
  \def\@@gls@checkquote{\@gls@checkquote#2""\null}%
  \else
  \def\@@gls@checkquote{\@gls@checkquote#2"#3\null}%
  \fi
  \fi
  \@@gls@checkquote
}
\def\@gls@checkescquote#1"#2"#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%

```

```

\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar
\@gls@quotechar\string"\@gls@quotechar}%
\def\@gls@checkescquote{\@gls@checkescquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar}%
\ifx\null#3\null
\def\@gls@checkescquote{\@gls@checkescquote#2\""\null}%
\else
\def\@gls@checkescquote{\@gls@checkescquote#2\#"#3\null}%
\fi
\fi
\@gls@checkescquote
}
\def\@gls@checkescactual#1\?#2\?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar
\@gls@quotechar\string"\@gls@actualchar}%
\def\@gls@checkescactual{\@gls@checkescactual#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar}%
\ifx\null#3\null
\def\@gls@checkescactual{\@gls@checkescactual#2\?\?\null}%
\else
\def\@gls@checkescactual{\@gls@checkescactual#2\?#3\null}%
\fi
\fi
\@gls@checkescactual
}
\def\@gls@checkescbar#1\|#2\|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null

```



```

\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkeschar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@encapchar
\@gls@quotechar\string"\@gls@encapchar}%
\def\@gls@checkeschar{\@gls@checkeschar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkeschar{\@gls@checkeschar#2\|\|\null}%
\else
\def\@gls@checkeschar{\@gls@checkeschar#2\|#3\null}%
\fi
\fi
\@gls@checkeschar
}
\def\@gls@checkeslevel#1\!#2\!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkeslevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@levelchar
\@gls@quotechar\string"\@gls@levelchar}%
\def\@gls@checkeslevel{\@gls@checkeslevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checkeslevel{\@gls@checkeslevel#2\!\!\null}%
\else
\def\@gls@checkeslevel{\@gls@checkeslevel#2\!#3\null}%
\fi
\fi
\@gls@checkeslevel
}
\def\@gls@checkbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%

```

```

\def\@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@gls@checkbar{\@gls@checkbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkbar{\@gls@checkbar#2|\null}%
\else
\def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}
\def\@gls@checklevel#1!#2!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checklevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
\def\@gls@checklevel{\@gls@checklevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checklevel{\@gls@checklevel#2!|\null}%
\else
\def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
\fi
\fi
\@gls@checklevel
}
\def\@gls@checkactual#1?#2?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%

```

```

        \def\@gls@checkactual{\@gls@checkactual#3\null}%
    \fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar}%
\ifx\null#3\null
\def\@gls@checkactual{\@gls@checkactual#2??\null}%
\else
\def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
\fi
\fi
\@gls@checkactual
}
\bgroup
\catcode'\0\relax
\catcode'\12\relax
|gdef|\@gls@checkesc#1\#2\#3\null{%
|\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
|\toks@={#1}%
\ifx\null#2\null
|\ifx\null#3\null
|\edef|\@gls@checkedmkidx{\the|\@gls@tmpb|\the|\toks@}%
|\def|\@gls@checkesc{\relax}%
\else
|\edef|\@gls@checkedmkidx{\the|\@gls@tmpb|\the|\toks@
|\@gls@quotechar|\backslashchar|\@gls@quotechar|\backslashchar}%
|\def|\@gls@checkesc{\@gls@checkesc#3\null}%
\fi
\else
|\edef|\@gls@checkedmkidx{\the|\@gls@tmpb|\the|\toks@
|\@gls@quotechar|\backslashchar}%
\ifx\null#3\null
|\def|\@gls@checkesc{\@gls@checkesc#2\\\null}%
\else
|\def|\@gls@checkesc{\@gls@checkesc#2\#3\null}%
\fi
\fi
|\@gls@checkesc
}
\egroup
\def\@gls@xdycheckquote#1"#2"#3\null{%
|\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
|\toks@={#1}%
\ifx\null#2\null
|\ifx\null#3\null
|\edef|\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
|\def\@gls@xdycheckquote{\relax}%
\else
|\edef|\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
|string"\string"}%

```

```

\def\@@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\string\}%
\ifx\null#3\null
\def\@@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
\else
\def\@@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
\fi
\fi
\fi
\@@gls@xdycheckquote
}
\edef\def@gls@xdycheckbackslash{%
\noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
##2\@backslashchar##3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter
{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={##1}%
\noexpand\ifx\noexpand\null##2\noexpand\null
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@@gls@xdycheckbackslash{\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
\noexpand\def\noexpand\@@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar}%
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\def\noexpand\@@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
\@backslashchar\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
##3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@@gls@xdycheckbackslash
}%
}
\def@gls@xdycheckbackslash

```

```

\newlength\gls@tmplen
\newcommand*\glsdohypertarget}[2]{%
  \glsdoshowtarget{#1}%
  {%
    \glsmeasureheight{\gls@tmplen}{#2}%
    \raisebox{\gls@tmplen}{\hypertarget{#1}{}}%
    #2%
  }%
}
\newcommand*\glsdohyperlink}[2]{%
  \glsdoshowtarget{#1}{\hyperlink{#1}{#2}}%
}
\newcommand*\glsdonohyperlink}[2]{#2}
\ifcsundef{hyperlink}%
{%
  \let\@glslink\glsdonohyperlink
}%
{%
  \let\@glslink\glsdohyperlink
}
\ifcsundef{hypertarget}%
{%
  \let\@glstarget\@secondoftwo
}%
{%
  \let\@glstarget\glsdohypertarget
}
\newcommand{\glsdisablehyper}{%
  \KV@glslink@hyperfalse
  \let\@glslink\glsdonohyperlink
  \let\@glstarget\@secondoftwo
}
\newcommand{\glsenablehyper}{%
  \KV@glslink@hypertrue
  \let\@glslink\glsdohyperlink
  \let\@glstarget\glsdohypertarget
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\newrobustcmd*\gls{\@gls@hyp@opt\@gls}
\newcommand*\@gls}[2][{}]{%
  \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2}[]}]%
}
\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\gls@ifplural\@secondoftwo
    \let\gls@scapscase\@firstofthree
    \let\gls@customtext\@empty
  }
}

```

```

\def\glsinsert{#3}%
\def\@glo@text{\csname gls@glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
\glsmfuaddmap{\gls}{\Gls}
\newcommand*{\@Gls}[2][{}]{%
\new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2}}[{}]}%
}
\def\@Gls@#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@secondoftwo
\let\gls@caps@case\@secondofthree
\let\gls@custom@text\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}
\glsmfublocker{\GLS}
\newcommand*{\@GLS}[2][{}]{%
\new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2}}[{}]}%
}
\def\@GLS@#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@secondoftwo
\let\gls@caps@case\@thirdofthree
\let\gls@custom@text\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\glspl}{\@gls@hyp@opt\@glspl}
\newcommand*{\@glspl}[2][{}]{%
\new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2}}[{}]}%
}
\def\@glspl@#1#2[#3]{%

```

```

\glsdoifexists{#2}%
{%
  \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
  \let\glsifplural\@firstoftwo
  \let\glsescapscase\@firstofthree
  \let\glscustomtext\@empty
  \def\glsinsert{#3}%
  \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
  \@gls@link[#1]{#2}{\@glo@text}%
  \@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
\glsmfuaddmap{\glspl}{\Glspl}
\newcommand*{\@Glspl}[2][{}]{%
  \new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2}[]}%
}
\def\@Glspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glsescapscase\@secondofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \@gls@do@glsunset{#2}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLSp1}{\@gls@hyp@opt\@GLSp1}
\glsmfublocker{\GLSp1}
\newcommand*{\@GLSp1}[2][{}]{%
  \new@ifnextchar[{\@GLSp1@{#1}{#2}}{\@GLSp1@{#1}{#2}[]}%
}
\def\@GLSp1@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glsescapscase\@thirdofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \@gls@do@glsunset{#2}%
  }%
  \glspostlinkhook
}

```

```

}
\newrobustcmd*{\glsdisp}{\@gls@hyp@opt\@glsdisp}
\newcommand*{\@glsdisp}[3] []{%
  \glsdoifexists{#2}{%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\gls@scaps\@firstofthree
    \def\gls@customtext{#3}%
    \def\gls@insert{}%
    \def\@glo@text{\csname gls@\gls@type @entryfmt@endcsname}%
    \@gls@link{#1}{#2}{\@glo@text}%
    \@gls@do@gls@unset{#2}%
  }%
  \gls@postlinkhook
}
\newrobustcmd*{\Glsdisp}{\@gls@hyp@opt\@Glsdisp}
\newcommand*{\@Glsdisp}[3] []{%
  \@glsdisp{#1}{#2}{\gls@sentencecase{#3}}%
}
\gls@fu@addmap{\glsdisp}{\Glsdisp}
\newcommand*{\@gls@link@nocheckfirsthyper}{}
\newcommand{\@gls@field@link}[3] {%
  \glsdoifexists{#2}{%
    {%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
      \@gls@link{#1}{#2}{#3}%
    }%
  }%
  \gls@postlinkhook
}
\newrobustcmd*{\gls@text}{\@gls@hyp@opt\@gls@text}
\newcommand*{\@gls@text}[2] []{%
  \new@ifnextchar[{\@gls@text@{#1}{#2}}{\@gls@text@{#1}{#2} []}]
\def\@gls@text@{#1}{#2}[#3] {%
  \@gls@field@link{#1}{#2}{\gls@entrytext{#2}{#3}}%
}
\newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
\gls@fu@blocker{\GLStext}
\newcommand*{\@GLStext}[2] []{%
  \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2} []}]
\def\@GLStext@{#1}{#2}[#3] {%
  \@gls@field@link{#1}{#2}{\gls@supercase{\gls@entrytext{#2}{#3}}}%
}
\newrobustcmd*{\Gls@text}{\@gls@hyp@opt\@Gls@text}
\gls@fu@addmap{\gls@text}{\Gls@text}
\newcommand*{\@Gls@text}[2] []{%
  \new@ifnextchar[{\@Gls@text@{#1}{#2}}{\@Gls@text@{#1}{#2} []}]
\def\@Gls@text@{#1}{#2}[#3] {%
  \@gls@field@link{#1}{#2}{\Gls@entrytext{#2}{#3}}%
}
\newrobustcmd*{\gls@first}{\@gls@hyp@opt\@gls@first}

```



```

\newcommand*{\@glsfirst}[2] [] {%
  \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2} []}]
\def\@glsfirst@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
}
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
\glsmfuaddmap{\glsfirst}{\Glsfirst}
\newcommand*{\@Glsfirst}[2] [] {%
  \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2} []}]
\def\@Glsfirst@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
\glsmfublocker{\GLSfirst}
\newcommand*{\@GLSfirst}[2] [] {%
  \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2} []}]
\def\@GLSfirst@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentryfirst{#2}#3}%
}
\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
\newcommand*{\@glsplural}[2] [] {%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} []}]
\def\@glsplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\glsmfuaddmap{\glsplural}{\Glsplural}
\newcommand*{\@Glsplural}[2] [] {%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} []}]
\def\@Glsplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
\newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
\glsmfublocker{\GLSplural}
\newcommand*{\@GLSplural}[2] [] {%
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} []}]
\def\@GLSplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{\@glsfirstplural}[2] [] {%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} []}]
\def\@glsfirstplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
\glsmfuaddmap{\glsfirstplural}{\Glsfirstplural}
\newcommand*{\@Glsfirstplural}[2] [] {%
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2} []}]
\def\@Glsfirstplural@#1#2[#3] {%

```

```

    \@gls@field@link{#1}#{2}{\glentryfirstplural{#2}#3}%
}
\newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
\glsmfublocker{\GLSfirstplural}
\newcommand*{\@GLSfirstplural}[2] [] {%
    \new@ifnextchar[{\@GLSfirstplural@{#1}#{2}}{\@GLSfirstplural@{#1}#{2} []}]
\def\@GLSfirstplural@#1#2[#3] {%
    \@gls@field@link{#1}#{2}{\glssupercase{\glentryfirstplural{#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}{\glentryname{#2}#3}%
}
\newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}
\glsmfuaddmap{\glsname}{\Glsname}
\newcommand*{\@Glsname}[2] [] {%
    \new@ifnextchar[{\@Glsname@{#1}#{2}}{\@Glsname@{#1}#{2} []}]
\def\@Glsname@#1#2[#3] {%
    \@gls@field@link{#1}#{2}{\glentryname{#2}#3}%
}
\newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}
\glsmfublocker{\GLSname}
\newcommand*{\@GLSname}[2] [] {%
    \new@ifnextchar[{\@GLSname@{#1}#{2}}{\@GLSname@{#1}#{2} []}]
\def\@GLSname@#1#2[#3] {%
    \@gls@field@link{#1}#{2}{\glssupercase{\glentryname{#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}{\glentrydesc{#2}#3}%
}
\newrobustcmd*{\Glsdesc}{\@gls@hyp@opt\@Glsdesc}
\glsmfuaddmap{\glsdesc}{\Glsdesc}
\newcommand*{\@Glsdesc}[2] [] {%
    \new@ifnextchar[{\@Glsdesc@{#1}#{2}}{\@Glsdesc@{#1}#{2} []}]
\def\@Glsdesc@#1#2[#3] {%
    \@gls@field@link{#1}#{2}{\glentrydesc{#2}#3}%
}
\newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\glsmfublocker{\GLSdesc}
\newcommand*{\@GLSdesc}[2] [] {%
    \new@ifnextchar[{\@GLSdesc@{#1}#{2}}{\@GLSdesc@{#1}#{2} []}]
\def\@GLSdesc@#1#2[#3] {%
    \@gls@field@link{#1}#{2}{\glssupercase{\glentrydesc{#2}#3}}%
}
\newrobustcmd*{\glsdescplural}{\@gls@hyp@opt\@glsdescplural}

```

```

\newcommand*{\@glsdescplural}[2] [] {%
  \new@ifnextchar[{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}]
\def\@glsdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\Glsdescplural}{\@gls@hyp@opt\@Glsdescplural}
\glsmfuaddmap{\glsdescplural}{\Glsdescplural}
\newcommand*{\@Glsdescplural}[2] [] {%
  \new@ifnextchar[{\@Glsdescplural@{#1}{#2}}{\@Glsdescplural@{#1}{#2} []}]
\def\@Glsdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\glsmfublocker{\GLSdescplural}
\newcommand*{\@GLSdescplural}[2] [] {%
  \new@ifnextchar[{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}]
\def\@GLSdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentrydescplural{#2}#3}}%
}
\newrobustcmd*{\glsymbol}{\@gls@hyp@opt\@glsymbol}
\newcommand*{\@glsymbol}[2] [] {%
  \new@ifnextchar[{\@glsymbol@{#1}{#2}}{\@glsymbol@{#1}{#2} []}]
\def\@glsymbol@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\GLSymbol}{\@gls@hyp@opt\@GLSymbol}
\glsmfublocker{\GLSymbol}
\newcommand*{\@GLSymbol}[2] [] {%
  \new@ifnextchar[{\@GLSymbol@{#1}{#2}}{\@GLSymbol@{#1}{#2} []}]
\def\@GLSymbol@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentrysymbol{#2}#3}}%
}
\newrobustcmd*{\glsymbolplural}{\@gls@hyp@opt\@glsymbolplural}
\newcommand*{\@glsymbolplural}[2] [] {%
  \new@ifnextchar[{\@glsymbolplural@{#1}{#2}}{\@glsymbolplural@{#1}{#2} []}]
\def\@glsymbolplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSymbolplural}{\@gls@hyp@opt\@GLSymbolplural}
\glsmfuaddmap{\glsymbolplural}{\GLSymbolplural}
\newcommand*{\@GLSymbolplural}[2] [] {%
  \new@ifnextchar[{\@GLSymbolplural@{#1}{#2}}{\@GLSymbolplural@{#1}{#2} []}]
\def\@GLSymbolplural@#1#2[#3] {%

```

```

    \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
\glsmfublocker{\GLSsymbolplural}
\newcommand*{\@GLSsymbolplural}[2] [] {%
    \new@ifnextchar[{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2} []}]
\def\@GLSsymbolplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrysymbolplural{#2}#3}}%
}
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
\newcommand*{\@glsuseri}[2] [] {%
    \new@ifnextchar[{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2} []}]
\def\@glsuseri@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryuseri{#2}#3}%
}
\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
\glsmfuaddmap{\glsuseri}{\Glsuseri}
\newcommand*{\@Glsuseri}[2] [] {%
    \new@ifnextchar[{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2} []}]
\def\@Glsuseri@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\glsmfublocker{\GLSuseri}
\newcommand*{\@GLSuseri}[2] [] {%
    \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2} []}]
\def\@GLSuseri@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseri{#2}#3}}%
}
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
\newcommand*{\@glsuserii}[2] [] {%
    \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2} []}]
\def\@glsuserii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryuserii{#2}#3}%
}
\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
\glsmfuaddmap{\glsuserii}{\Glsuserii}
\newcommand*{\@Glsuserii}[2] [] {%
    \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2} []}]
\def\@Glsuserii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\GLSuserii}{\@gls@hyp@opt\@GLSuserii}
\glsmfublocker{\GLSuserii}
\newcommand*{\@GLSuserii}[2] [] {%
    \new@ifnextchar[{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2} []}]
\def\@GLSuserii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuserii{#2}#3}}%
}
\newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}

```

```

\newcommand*{\@glsuseriii}[2] [] {%
  \new@ifnextchar[{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2} []}]
\def\@glsuseriii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
\glsmfuaddmap{\glsuseriii}{\Glsuseriii}
\newcommand*{\@Glsuseriii}[2] [] {%
  \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2} []}]
\def\@Glsuseriii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\GLSuseriii}{\@gls@hyp@opt\@GLSuseriii}
\glsmfublocker{\GLSuseriii}
\newcommand*{\@GLSuseriii}[2] [] {%
  \new@ifnextchar[{\@GLSuseriii@{#1}{#2}}{\@GLSuseriii@{#1}{#2} []}]
\def\@GLSuseriii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseriii{#2}#3}}%
}
\newrobustcmd*{\glsuseriv}{\@gls@hyp@opt\@glsuseriv}
\newcommand*{\@glsuseriv}[2] [] {%
  \new@ifnextchar[{\@glsuseriv@{#1}{#2}}{\@glsuseriv@{#1}{#2} []}]
\def\@glsuseriv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
\glsmfuaddmap{\glsuseriv}{\Glsuseriv}
\newcommand*{\@Glsuseriv}[2] [] {%
  \new@ifnextchar[{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2} []}]
\def\@Glsuseriv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\GLSuseriv}{\@gls@hyp@opt\@GLSuseriv}
\glsmfublocker{\GLSuseriv}
\newcommand*{\@GLSuseriv}[2] [] {%
  \new@ifnextchar[{\@GLSuseriv@{#1}{#2}}{\@GLSuseriv@{#1}{#2} []}]
\def\@GLSuseriv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseriv{#2}#3}}%
}
\newrobustcmd*{\glsuserv}{\@gls@hyp@opt\@glsuserv}
\newcommand*{\@glsuserv}[2] [] {%
  \new@ifnextchar[{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2} []}]
\def\@glsuserv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\Glsuserv}{\@gls@hyp@opt\@Glsuserv}
\glsmfuaddmap{\glsuserv}{\Glsuserv}
\newcommand*{\@Glsuserv}[2] [] {%
  \new@ifnextchar[{\@Glsuserv@{#1}{#2}}{\@Glsuserv@{#1}{#2} []}]
\def\@Glsuserv@#1#2[#3] {%

```

```

    \@gls@field@link{#1}{#2}{\glsentryuser{#2}#3}%
}
\newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
\glsmfublocker{\GLSuserv}
\newcommand*{\@GLSuserv}[2][\%
\new@ifnextchar[{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2}[]}]
\def\@GLSuserv@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuser{#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}{\glsentryuser{#2}#3}%
}
\newrobustcmd*{\Glsuservi}{\@gls@hyp@opt\@Glsuservi}
\glsmfuaddmap{\glsuservi}{\Glsuservi}
\newcommand*{\@Glsuservi}[2][\%
    \new@ifnextchar[{\@Glsuservi@{#1}{#2}}{\@Glsuservi@{#1}{#2}[]}]
\def\@Glsuservi@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryuser{#2}#3}%
}
\newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
\glsmfublocker{\GLSuservi}
\newcommand*{\@GLSuservi}[2][\%
    \new@ifnextchar[{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2}[]}]
\def\@GLSuservi@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuser{#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\glsescapscase\@firstofthree
        \let\glsinsert\@empty
        \def\glscustomtext{%
            \acronymfont{\glsentryshort{#2}}#3%
        }%
        \@gls@link[#1]{#2}{\csname gls@\gls@type @entryfmt\endcsname}%
    }%
    \glspostlinkhook
}
\newrobustcmd*{\Acrshort}{\@gls@hyp@opt\ns@Acrshort}
\glsmfuaddmap{\Acrshort}{\Acrshort}
\newcommand*{\ns@Acrshort}[2][\%

```

```

\new@ifnextchar [{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2} []}%
}
\def\@Acrshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@secondoftwo
\let\glscapscase\@secondofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\Glsentryshort{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\ACRshort}{\@gl@hyp@opt\ns@ACRshort}
\glsmfublocker{\ACRshort}
\newcommand*{\ns@ACRshort}[2] []{%
\new@ifnextchar [{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}%
}
\def\@ACRshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\glsuppercase{\acronymfont{\glsentryshort{#2}}#3}%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\acrshortpl}{\@gl@hyp@opt\ns@acrshortpl}
\newcommand*{\ns@acrshortpl}[2] []{%
\new@ifnextchar [{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} []}%
}
\def\@acrshortpl#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%

```

```

        \acronymfont{\glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls@glstype @entryfmt\endcsname}%
}
\glspostlinkhook
}
\newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\ns@Acrshortpl}
\glsmfuaddmap{\Acrshortpl}{\Acrshortpl}
\newcommand*{\ns@Acrshortpl}[2] []{%
    \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2} []}%
}
\def\@Acrshortpl#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \def\glslabel{#2}%
        \let\glsifplural\@firstoftwo
        \let\glsescapscase\@secondofthree
        \let\glsinsert\@empty
        \def\glscustomtext{%
            \acronymfont{\Glsentryshortpl{#2}}#3%
        }%
        \@gls@link[#1]{#2}{\csname gls@\gls@glstype @entryfmt\endcsname}%
    }%
    \glspostlinkhook
}
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\ns@ACRshortpl}
\glsmfublocker{\ACRshortpl}
\newcommand*{\ns@ACRshortpl}[2] []{%
    \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2} []}%
}
\def\@ACRshortpl#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \def\glslabel{#2}%
        \let\glsifplural\@firstoftwo
        \let\glsescapscase\@thirdofthree
        \let\glsinsert\@empty
        \def\glscustomtext{%
            \glsuppercase{\acronymfont{\glsentryshortpl{#2}}#3}%
        }%
        \@gls@link[#1]{#2}{\csname gls@\gls@glstype @entryfmt\endcsname}%
    }%
    \glspostlinkhook
}
\newrobustcmd*{\acrlong}{\@gls@hyp@opt\ns@acrlong}
\newcommand*{\ns@acrlong}[2] []{%
    \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2} []}%
}

```



```

\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsentrylong{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls@type @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrlong}{\@gls@hyp@opt\@ns@Acrlong}
\glsmfuaddmap{\Acrlong}{\Acrlong}
\newcommand*{\ns@Acrlong}[2][]{%
  \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2}[]}]%
}
\def\@Acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsapsaps\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylong{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls@type @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRlong}{\@gls@hyp@opt\@ns@ACRlong}
\glsmfublocker{\ACRlong}
\newcommand*{\ns@ACRlong}[2][]{%
  \new@ifnextchar[{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2}[]}]%
}
\def\@ACRlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsapsaps\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsuppercase{\glsentrylong{#2}#3}%
    }%
  }%
}

```

```

    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\ns@acrlongpl}
\newcommand*{\ns@acrlongpl}[2][{}]{%
  \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2}[]}%
}
\def\@acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\AcrLongpl}{\@gls@hyp@opt\ns@AcrLongpl}
\glsmfuaddmap{\AcrLongpl}{\AcrLongpl}
\newcommand*{\ns@AcrLongpl}[2][{}]{%
  \new@ifnextchar[{\@AcrLongpl{#1}{#2}}{\@AcrLongpl{#1}{#2}[]}%
}
\def\@AcrLongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glsapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRLongpl}{\@gls@hyp@opt\ns@ACRLongpl}
\glsmfublocker{\ACRLongpl}
\newcommand*{\ns@ACRLongpl}[2][{}]{%
  \new@ifnextchar[{\@ACRLongpl{#1}{#2}}{\@ACRLongpl{#1}{#2}[]}%
}
\def\@ACRLongpl#1#2[#3]{%

```

```

\glsdoifexists{#2}%
{%
  \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
  \def\glslabel{#2}%
  \let\glsifplural\@firstoftwo
  \let\glsescapscase\@thirdofthree
  \let\glsinsert\@empty
  \def\glscustomtext{%
    \glsuppercase{\glsentrylongpl{#2}#3}%
  }%
  \@gls@link[#1]{#2}{\csname gls@\gls@type @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newcommand*{\@gls@entry@field}[2]{%
  \csuse{glo@\glsdetoklabel{#1}@#2}%
}
\newcommand*{\glsletentryfield}[3]{%
  \letcs{#1}{glo@\glsdetoklabel{#2}@#3}%
}
\newrobustcmd*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@#2}%
    \ifdef\@glo@text
    {%
      \expandafter\glsentencecase\expandafter{\@glo@text}%
    }%
    {%
      ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
        entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
        label and the field name}%
    }%
  }%
  }%
  }%
  }%
}
\newcommand{\@Glsentryfield}[2]{%
  \glsstexorpdfstring{\@Gls@entry@field{#1}{#2}}%
  {\MFUsentencecase{\@gls@entry@field{#1}{#2}}}%
}
\ifdef\texorpdfstring
{\newcommand{\glsstexorpdfstring}{\texorpdfstring}}
{\newcommand{\glsstexorpdfstring}[2]{#1}}
\newcommand*{\glsentryname}[1]{\@gls@entry@field{#1}{name}}
\newcommand*{\Glsentryname}[1]{%
  \glsstexorpdfstring{\protect\@Gls@entryname{#1}}%
  {\MFUsentencecase{\@gls@entry@field{#1}{name}}}%
}

```

```

\glsmfuaddmap{\glsentryname}{\Glsentryname}
\newcommand*{\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
\newrobustcmd*{\@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \protected@edef\@glo@text{\@Gls@entry@field{#1}{name}}%
    \expandafter\glsentencecase\expandafter{\@glo@text}%
  }%
  {%
    \@Gls@entry@field{#1}{name}%
  }%
}
\newcommand*{\@org@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo\@glsdetoklabel{#1}@name}%
    \expandafter\@gls@getbody\@glo@text{\@nil
    \expandafter\ifx\@gls@body\glsentrylong\relax
    \expandafter\Glsentrylong\@gls@rest
  \else
    \expandafter\ifx\@gls@body\glsentryshort\relax
    \expandafter\Glsentryshort\@gls@rest
  \else
    \expandafter\ifx\@gls@body\acronymfont\relax
    {%
      \let\glsentryshort\Glsentryshort
      \@glo@text
    }%
  \else
    \expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
    {%
      \let\glsentryshort\Glsentryshort
      \@glo@text
    }%
  \else
    \expandafter\glsentencecase\expandafter{\@glo@text}%
  \fi
  \fi
  \fi
  \fi
  }%
  {%
    \@Gls@entry@field{#1}{name}%
  }%
}
\newcommand*{\glsentrydesc}[1]{\@Gls@entry@field{#1}{desc}}
\newcommand*{\Glsentrydesc}[1]{\@Gls@entryfield{#1}{desc}}
\glsmfuaddmap{\glsentrydesc}{\Glsentrydesc}

```

```

\newcommand*\glentrydescplural}[1]{%
  \@gls@entry@field{#1}{descplural}%
}
\newcommand*\Glsentrydescplural}[1]{%
  \@Glsentryfield{#1}{descplural}%
}
\glsmfuaddmap{\glentrydescplural}{\Glsentrydescplural}
\newcommand*\glentrytext}[1]{\@gls@entry@field{#1}{text}}
\newcommand*\Glsentrytext}[1]{%
  \@Glsentryfield{#1}{text}%
}
\glsmfuaddmap{\glentrytext}{\Glsentrytext}
\newcommand*\glentryplural}[1]{%
  \@gls@entry@field{#1}{plural}%
}
\newcommand*\Glsentryplural}[1]{%
  \@Glsentryfield{#1}{plural}%
}
\glsmfuaddmap{\glentryplural}{\Glsentryplural}
\newcommand*\glentrysymbol}[1]{%
  \@gls@entry@field{#1}{symbol}%
}
\newcommand*\Glsentrysymbol}[1]{%
  \@Glsentryfield{#1}{symbol}%
}
\glsmfuaddmap{\glentrysymbol}{\Glsentrysymbol}
\newcommand*\glentrysymbolplural}[1]{%
  \@gls@entry@field{#1}{symbolplural}%
}
\newcommand*\Glsentrysymbolplural}[1]{%
  \@Glsentryfield{#1}{symbolplural}%
}
\glsmfuaddmap{\glentrysymbolplural}{\Glsentrysymbolplural}
\newcommand*\glentryfirst}[1]{%
  \@gls@entry@field{#1}{first}%
}
\newcommand*\Glsentryfirst}[1]{%
  \@Glsentryfield{#1}{first}%
}
\glsmfuaddmap{\glentryfirst}{\Glsentryfirst}
\newcommand*\glentryfirstplural}[1]{%
  \@gls@entry@field{#1}{firstpl}%
}
\newcommand*\Glsentryfirstplural}[1]{%
  \@Glsentryfield{#1}{firstpl}%
}
\glsmfuaddmap{\glentryfirstplural}{\Glsentryfirstplural}
\newcommand*\glscapitalisewords}[1]{%
  \capitalisewords{#1}%
}

```

```

\newcommand*\glsentrytitlecase}[2]{%
  \glstexorpdfstring
  {\@glsentrytitlecase{#1}{#2}}%
  {\MFUsentencecase{\@gls@entry@field{#1}{#2}}}%
}
\newrobustcmd*\@glsentrytitlecase}[2]{%
  \glsdoifexists{#1}%
  {%
    \glsfieldfetch{#1}{#2}{\@gls@value}%
    \expandafter\glscapitalisewords\expandafter{\@gls@value}%
  }%
}
\newcommand*\glsentrytype}[1]{\@gls@entry@field{#1}{type}}
\newcommand*\glsentrysort}[1]{%
  \@gls@entry@field{#1}{sort}%
}
\newcommand*\glsentryparent}[1]{%
  \@gls@entry@field{#1}{parent}%
}
\newcommand*\glsentryuseri}[1]{%
  \@gls@entry@field{#1}{useri}%
}
\newcommand*\Glsentryuseri}[1]{%
  \@Glsentryfield{#1}{useri}%
}
\glsmfuaddmap{\glsentryuseri}{\Glsentryuseri}
\newcommand*\glsentryuserii}[1]{%
  \@gls@entry@field{#1}{userii}%
}
\newcommand*\Glsentryuserii}[1]{%
  \@Glsentryfield{#1}{userii}%
}
\glsmfuaddmap{\glsentryuserii}{\Glsentryuserii}
\newcommand*\glsentryuseriii}[1]{%
  \@gls@entry@field{#1}{useriii}%
}
\newcommand*\Glsentryuseriii}[1]{%
  \@Glsentryfield{#1}{useriii}%
}
\glsmfuaddmap{\glsentryuseriii}{\Glsentryuseriii}
\newcommand*\glsentryuseriv}[1]{%
  \@gls@entry@field{#1}{useriv}%
}
\newcommand*\Glsentryuseriv}[1]{%
  \@Glsentryfield{#1}{useriv}%
}
\glsmfuaddmap{\glsentryuseriv}{\Glsentryuseriv}
\newcommand*\glsentryuserv}[1]{%
  \@gls@entry@field{#1}{userv}%
}
}

```

```

\newcommand*{\Glsentryuser}[1]{%
  \@Glsentryfield{#1}{user}%
}
\glsmfuaddmap{\glsentryuser}{\Glsentryuser}
\newcommand*{\glsentryuseri}[1]{%
  \@gls@entry@field{#1}{useri}%
}
\newcommand*{\Glsentryuseri}[1]{%
  \@Glsentryfield{#1}{useri}%
}
\glsmfuaddmap{\glsentryuseri}{\Glsentryuseri}
\newcommand*{\glsentryshort}[1]{\@gls@entry@field{#1}{short}}
\newcommand*{\Glsentryshort}[1]{%
  \@Glsentryfield{#1}{short}%
}
\glsmfuaddmap{\glsentryshort}{\Glsentryshort}
\newcommand*{\glsentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}
\newcommand*{\Glsentryshortpl}[1]{%
  \@Glsentryfield{#1}{shortpl}%
}
\glsmfuaddmap{\glsentryshortpl}{\Glsentryshortpl}
\newcommand*{\glsentrylong}[1]{\@gls@entry@field{#1}{long}}
\newcommand*{\Glsentrylong}[1]{%
  \@Glsentryfield{#1}{long}%
}
\glsmfuaddmap{\glsentrylong}{\Glsentrylong}
\newcommand*{\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}
\newcommand*{\Glsentrylongpl}[1]{%
  \@Glsentryfield{#1}{longpl}%
}
\glsmfuaddmap{\glsentrylongpl}{\Glsentrylongpl}
\newcommand*{\glsentryfull}[1]{%
  \glsentrylong{#1} (\glsentryshort{#1})%
}
\newcommand*{\Glsentryfull}[1]{%
  \Glsentrylong{#1} (\Glsentryshort{#1})%
}
\glsmfuaddmap{\glsentryfull}{\Glsentryfull}
\newcommand*{\glsentryfullpl}[1]{%
  \acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newcommand*{\Glsentryfullpl}[1]{%
  \Glsentrylongpl{#1} (\glsentryshortpl{#1})%
}
\glsmfuaddmap{\glsentryfullpl}{\Glsentryfullpl}
\newcommand*{\glsentrynumberlist}[1]{%
  \glsdoifexists{#1}%
  {%
    \@gls@entry@field{#1}{numberlist}%
  }%
}

```

```

}
\@ifpackageloaded{hyperref} {%
  \newcommand*{\glsdisplaynumberlist}[1]{%
    \GlossariesWarning
    {%
      \string\glsdisplaynumberlist\space
      doesn't work with hyperref.^^JUsing
      \string\glsentrynumberlist\space instead%
    }%
    \glsentrynumberlist{#1}%
  }%
}%
{%
  \newcommand*{\glsdisplaynumberlist}[1]{%
    \glsdoifexists{#1}%
    {%
      \bgroup
      \protected@edef\@glo@label{\glsdetoklabel{#1}}%
      \let\@org@glsnumberformat\glsnumberformat
      \def\glsnumberformat##1{##1}%
      \protected@edef\the@numberlist{%
        \csname glo@\@glo@label @numberlist\endcsname}%
      \def\@gls@numlist@sep{}%
      \def\@gls@numlist@nextsep{}%
      \def\@gls@numlist@lastsep{}%
      \def\@gls@thislist{}%
      \def\@gls@donext@def{}%
      \renewcommand\do[1]{%
        \protected@edef\@gls@thislist{%
          \@gls@thislist
          \noexpand\@gls@numlist@sep
          ##1%
        }%
        \let\@gls@numlist@sep\@gls@numlist@nextsep
        \def\@gls@numlist@nextsep{\glsnumlistsep}%
        \@gls@donext@def
        \def\@gls@donext@def{%
          \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
        }%
      }%
      \expandafter \glsnumlistparser \expandafter{\the@numberlist}%
      \let\@gls@numlist@sep\@gls@numlist@lastsep
      \@gls@thislist
    \egroup
  }%
}
}
\newcommand*{\glsnumlistsep}{, }
\newcommand*{\glsnumlistlastsep}{ \& }
\newcommand*{\gls hyperlink}[2][\glsentrytext{\@glo@label}]{%

```



```

\def\@glo@label{#2}%
\@glslink{\glo@linkprefix\glsdetoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*\glsadd[2] [] {%
  \@gls@adjustmode
  \glsdoifexists{#2}%
  {%
    \def\@glsnumberformat{glsnumberformat}%
    \protected@edef\@gls@counter{\csname glo@glsdetoklabel{#2}@counter\endcsname}%
    \setkeys{glossadd}{#1}%
    \@gls@saveentrycounter
    \@gls@setsort{#2}%
    \@do@wrglossary{#2}%
  }%
}
\glsmfuexcl{\glsadd}
\newcommand*\@gls@adjustmode{}
\AtBeginDocument{\renewcommand*\@gls@adjustmode{\ifvmode\mbox{}\fi}}
\newrobustcmd*\glsaddall[1] [] {%
  \protected@edef\@glo@type{\@glo@types}%
  \setkeys{glossadd}{#1}%
  \forallglsentries[\@glo@type]{\@glo@entry}{%
    \glsadd[#1]{\@glo@entry}%
  }%
}
\newrobustcmd*\glsaddallunused[1][\@glo@types]{%
  \forallglsentries[#1]{\@glo@entry}%
  {%
    \ifglsused{\@glo@entry}{\glsadd[format=glsignore]{\@glo@entry}}%
  }%
}
\newcommand*\glsignore[1]{}
\edef\glsopenbrace{\expandafter@gobble\string\{ }
\edef\glsclosebrace{\expandafter@gobble\string\} }
\edef\glsbackslash{\expandafter@gobble\string\ }
\edef\glsquote#1{\string"#1\string"}
\edef\glspercentchar{\expandafter@gobble\string\% }
\edef\glsstildechar{\string~ }
\ifglsxindy
  \newcommand*\@glsfirstletter}{A}
\fi
\newcommand*\GlsSetXdyFirstLetterAfterDigits{%
  \@ifstar\s@GlsSetXdyFirstLetterAfterDigits@GlsSetXdyFirstLetterAfterDigits}
\ifglsxindy
  \newcommand*\@GlsSetXdyFirstLetterAfterDigits[1]{%
    \renewcommand*\@glsfirstletter}{#1}}
  \newcommand*\s@GlsSetXdyFirstLetterAfterDigits[1]{%
    \renewcommand*\@glsfirstletter}{#1}}

```

```

    \@onelevel@sanitize\@glsfirstletter
  }
\else
  \newcommand*\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \glsnoxywarning\GlsSetXdyFirstLetterAfterDigits}
  \newcommand*\s@GlsSetXdyFirstLetterAfterDigits}{%
    \@GlsSetXdyFirstLetterAfterDigits
  }
\fi
\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]{%
    \glsnoxywarning\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 \@glsminrange^^J%
        )
    }%
}%
}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
    :unverified )}%
\write\glswrite{(markup-crossref-list
    :class \string"see\string"^^J\space\space\space
    :open \string"\string\glsseeformat\string"

```

```

        :close \string"{}\string"}}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
  :open \string"\string
  \glossarysection[\string\glossarytoctitle]{\string
  \glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
  {%
    \@for\@this@attr:=\@xdyattributelist\do{%
      \protected@write\glswrite{ }\string\providecommand*%
      \expandafter\string
      \csname glsX\@this@ctr X\@this@attr\endcsname[2]%
      {%
        \string\setentrycounter
        [\expandafter\@gobble\string\#1]{\@this@ctr}%
        \expandafter\string
        \csname\@this@attr\endcsname
        {\expandafter\@gobble\string\#2}%
      }%
    }%
  }%
}%
\write\glswrite{%
  \string\begin
  {theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
  \space\space:close \string"\glspercentchar\glstildechar n\string
  \end{theglossary}\string\glossarypostamble
  \glstildechar n\string" ^^J\space\space\space
  :tree}}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \glstildechar n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list
  :sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
  :sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffiXFF
\@onelevel@sanitize\gls@suffiXFF

```

```

\ifx\gls@suffiF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffiF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffiFF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffiFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
  :open-head \string"\string\glsgroupheading
  \glsopenbrace\string"^^J\space\space\space
  :close-head \string"\glsclosebrace\string")}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}%
\write\glswrite{\@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}%
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax
\ifundef{\glswrite}{\newwrite\glswrite}{}\relax
\openout\glswrite=\istfilename
\write\glswrite{\glspercentchar\space makeindex style file
  created by the glossaries package}
\write\glswrite{\glspercentchar\space for document
  '\jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\@gls@actualchar'}
\write\glswrite{encap '\@gls@encapchar'}
\write\glswrite{level '\@gls@levelchar'}
\write\glswrite{quote '\@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
  \glossarytoctitle]{\string\glossarytitle}\string
  \glossarypreamble\string\n\string\begin{theglossary}\string

```

```

    \glossaryheader\string\n\string"}
\write\glswrite{postamble \string"\string%\string\n\string
  \end{theglossary}\string\glossarypostamble\string\n
  \string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
  \string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}
\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
  \string"\string\relax \string\glsresetentrylist\string\n
  \string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
  \string"\string\relax \string\glsresetentrylist\string\n
  \string"}
\write\glswrite{delim_0 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_2 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_t \string"\string\}\string}\string"}
\write\glswrite{delim_n \string"\string\delimN \string"}
\write\glswrite{delim_r \string"\string\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
  \string"\string}\string\relax
  \string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"glscpositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF@empty
\else
  \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}

```

```

\fi
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremake\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{}
\ifglsxindy
\newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\newcommand*\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}%
\ifpackageloaded{tracklang}%
{%
\IfTrackedLanguage{german}%
{%
\def\@gls@extramakeindexopts{-g}%
}%
}%
}%
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkquote####1###2#1###3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={###1}%
\noexpand\ifx\noexpand\null####2\noexpand\null
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####2#1#1\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####2#1###3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%

```

```

\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
  \noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
    \noexpand\def\noexpand\@gls@checkedmkidx{%
      \noexpand\expandafter\noexpand\@gls@checkquote####1\noexpand\@nil
      #1#1\noexpand\null
    }
    \noexpand\expandafter\noexpand\@gls@updatechecked
    \noexpand\@gls@checkedmkidx{####1}%
  }
  \noexpand\def\noexpand\@gls@checkedmkidx{%
    \noexpand\expandafter\noexpand\@gls@checkescquote####1\noexpand\@nil
    \expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
    \noexpand\null
  }
  \noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{####1}%
  \noexpand\def\noexpand\@gls@checkedmkidx{%
    \noexpand\expandafter\noexpand\@gls@checkescactual####1\noexpand\@nil
    \noexpand\?\noexpand\?\noexpand\null
  }
  \noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{####1}%
  \noexpand\def\noexpand\@gls@checkedmkidx{%
    \noexpand\expandafter\noexpand\@gls@checkactual####1\noexpand\@nil
    \noexpand?\noexpand?\noexpand\null
  }
  \noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{####1}%
  \noexpand\def\noexpand\@gls@checkedmkidx{%
    \noexpand\expandafter\noexpand\@gls@checkbar####1\noexpand\@nil
    \noexpand|\noexpand|\noexpand\null
  }
  \noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{####1}%
  \noexpand\def\noexpand\@gls@checkedmkidx{%
    \noexpand\expandafter\noexpand\@gls@checkescbar####1\noexpand\@nil
    \noexpand\|\noexpand\|\noexpand\null
  }
  \noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{####1}%
  \noexpand\def\noexpand\@gls@checkedmkidx{%
    \noexpand\expandafter\noexpand\@gls@checklevel####1\noexpand\@nil
    \noexpand!\noexpand!\noexpand\null
  }
  \noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{####1}%
  \noexpand\def\noexpand\@gls@checkedmkidx{%
    \noexpand\expandafter\noexpand\@gls@checkesclevel####1\noexpand\@nil
    \noexpand\!\noexpand\!\noexpand\null
  }
  \noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{####1}%
  \noexpand\def\noexpand\@gls@checkedmkidx{%
    \noexpand\expandafter\noexpand\@gls@checkesc####1\noexpand\@nil
    \@backslashchar\@backslashchar\noexpand\null
  }
  \noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{####1}%
}%

```



```

}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
  \noexpand\def\noexpand\@gls@checkescquote####1%
    \expandonce{\csname#1\endcsname}####2\expandonce{\csname#1\endcsname}%
    ####3\noexpand\null{%
      \noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
      \noexpand\toks@={####1}%
      \noexpand\ifx\noexpand\null####2\noexpand\null
        \noexpand\ifx\noexpand\null####3\noexpand\null
          \noexpand\edef\noexpand\@gls@checkedmkidx{%
            \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
          \noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
          \noexpand\else
            \noexpand\edef\noexpand\@gls@checkedmkidx{%
              \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
              \noexpand\@gls@quotechar\noexpand\string\expandonce{%
                \csname#1\endcsname}\noexpand\@gls@quotechar
                \noexpand\@gls@quotechar\noexpand\string\expandonce{%
                  \csname#1\endcsname}\noexpand\@gls@quotechar}%
              \noexpand\def\noexpand\@gls@checkescquote{%
                \noexpand\@gls@checkescquote####3\noexpand\null}%
            \noexpand\fi
          \noexpand\else
            \noexpand\edef\noexpand\@gls@checkedmkidx{%
              \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
              \noexpand\@gls@quotechar\noexpand\string
                \expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
            \noexpand\ifx\noexpand\null####3\noexpand\null
              \noexpand\def\noexpand\@gls@checkescquote{%
                \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
                \expandonce{\csname#1\endcsname}\noexpand\null}%
            \noexpand\else
              \noexpand\def\noexpand\@gls@checkescquote{%
                \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
                ####3\noexpand\null}%
            \noexpand\fi
          \noexpand\fi
        \noexpand\@gls@checkescquote
      }%
    }%
  }%
\@gls@docheckquotedef
}
\newcommand*\@gls@nosetquote}[1]{\PackageError{glossaries}%
  {\string\GlsSetQuote\space not permitted here}%
  {\Move \string\GlsSetQuote\space earlier in the preamble, as
  soon as possible after glossaries.sty has been loaded}}
\fi
\newcommand*\@gls@extramakeindexopts}[1]{
\newcommand{\noist}{%

```

```

\@gls@addpredefinedattributes
\let\writeist\relax
}
\newcommand{\@gls@noist@toolate}{%
\GlossariesWarning{too late to use \string\noist. The style file
has already been written}%
}
\newcommand*{\@makeglossary}[1]{%
\ifglossaryexists{#1}%
{%
\ifglssavewrites
\expandafter\newtoks\csname glo@#1@filetok\endcsname
\else
\expandafter\newwrite\csname glo@#1@file\endcsname
\expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
\fi
\@gls@renewglossary
\writeist
}%
{%
\PackageError{glossaries}%
{Glossary type ‘#1’ not defined}%
{New glossaries must be defined before using \string\makeglossaries}%
}%
}
\newcommand*{\@glsopenfile}[2]{%
\immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
\PackageInfo{glossaries}{Writing glossary file
\jobname.\csname @glotype@#2@out\endcsname}%
}
\newcommand*{\@closegls}[1]{%
\closeout\csname glo@#1@file\endcsname
}
\ifglsexindy
\newcommand*{\@gls@automake}[1]{%
\ifglossaryexists{#1}
{%
\@closegls{#1}%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-M ord/letorder }}%
{\let\@gls@order\@empty}%
\gls@xdy@set@lang{#1}%
\ifdefvoid\gls@codepage
{\let\@gls@codepage\@empty}%
{\def\@gls@codepage{-C \gls@codepage\space}}%
\edef\@gls@dothiswrite{\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \gls@istfilebase\space

```

```

        \@gls@codepage
        -t \jobname.\csuse{@glotype@#1@log}
        -o \jobname.\csuse{@glotype@#1@in}
        \jobname.\csuse{@glotype@#1@out}}%
    }%
    \@gls@dothiswrite
}
{
    \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}
}
\else
\newcommand*{\@gls@automake}[1]{%
    \ifglossaryexists{#1}
    {%
        \@closegls{#1}%
        \ifdefstring{\glsorder}{letter}%
        {\def\@gls@order{-1 }}%
        {\let\@gls@order\@empty}%
        \edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
            -s \listfilename\space
            -t \jobname.\csuse{@glotype@#1@log}
            -o \jobname.\csuse{@glotype@#1@in}
            \jobname.\csuse{@glotype@#1@out}}}%
        }%
        \@gls@dothiswrite
    }%
    {
        \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
    }%
}
\fi
\ifglsxindy
\newcommand*{\@gls@automake@immediate}[1]{%
    \ifglossaryexists{#1}
    {%
        \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
        {%
            \ifdefstring{\glsorder}{letter}%
            {\def\@gls@order{-M ord/letorder }}%
            {\let\@gls@order\@empty}%
            \gls@xdy@set@lang{#1}%
            \ifdefvoid\gls@codepage
            {\let\@gls@codepage\@empty}%
            {\def\@gls@codepage{-C \gls@codepage\space}}%
            \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
                -I xindy
                \@gls@order
                -L \@gls@langmod\space
                -M \gls@istfilebase\space

```

```

        \@gls@codepage
        -t \jobname.\csuse{@glotype@#1@log}
        -o \jobname.\csuse{@glotype@#1@in}
        \jobname.\csuse{@glotype@#1@out}}%
    }%
    \@gls@dothiswrite
}
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{
    \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
    \ifglossaryexists{#1}
    {%
        \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
        {%
            \ifdefstring{\glsorder}{letter}%
            {\def\@gls@order{-l }}%
            {\let\@gls@order\@empty}%
            \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order
            -s \istfilename\space
            -t \jobname.\csuse{@glotype@#1@log}
            -o \jobname.\csuse{@glotype@#1@in}
            \jobname.\csuse{@glotype@#1@out}}%
            }%
            \@gls@dothiswrite
        }%
        {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
        doesn't exist. Rerun may be required}}%
    }%
    {
        \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
    }%
}
}
\fi
\newcommand*{\@warn@nomakeglossaries}{}
\newcommand*{\@warn@nomakeglossaries}{\@warn@nomakeglossaries}
\newcommand{\@gls@automake@types}{\@glo@types}
\newcommand{\@gls@automake@immediate}{%
    \ifnum\gls@automake@nr=2\relax
    \protected@edef\gls@tmp{\@gls@automake@types}%
    \@for\@gls@type:=\gls@tmp\do{%
        \ifdefempty{\@gls@type}{%
            {\@gls@automake@immediate{\@gls@type}}%
        }%
    }%
}
\glsautomakefalse

```

```

\renewcommand*{\@gls@doautomake}{}%
\else
\ifdefempty\gls@automake@makegloss{%
{%
\IfFileExists{\jobname.aux}%
{\immediate\write18{\gls@automake@makegloss\space \jobname}}{%
\glsautomakefalse
\renewcommand*{\@gls@doautomake}{}%
}%
\fi
}
\newcommand*{\makeglossaries}{%
\ifglxindy
\def\glsindexingsetting{xindy}%
\else
\def\glsindexingsetting{makeindex}%
\fi
\@domakeglossaries
{%
\@gls@@automake@immediate
\ifundef\glswrite{\newwrite\glswrite}{%
\protected@write\@auxout{\string\providecommand\string\@glsorder[1]{}}
\protected@write\@auxout{\string\providecommand\string\@istfilename[1]{}}
\ifundef\@gls@extramakeindexopts
}{%
{%
\protected@write\@auxout{\string\providecommand
\string\@gls@extramakeindexopts[1]{}}
\protected@write\@auxout{\string\@gls@extramakeindexopts
{\@gls@extramakeindexopts}}%
}%
\protected@write\@auxout{\string\@istfilename{\istfilename}}%
\protected@write\@auxout{\string\@glsorder{\glsorder}}
\@for\@glo@type:=\@glo@types\do{%
\ifthenelse{\equal{\@glo@type}{}}{}}{%
\@makeglossary{\@glo@type}}%
}%
\renewcommand*\newglossary[4][{}]{%
\PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%
\let\@makeglossary\@gobble
\let\makeglossaries\relax
\@disable@onlypremakeg
\let\gls@checkseeallowed\relax
\let\warn@nomakeglossaries\relax
\let\warn@noprintglossary\@makeglossaries@warn@noprintglossary
\ifglssavenumberlist
\edef\@gls@dodolistparser{\noexpand\DeclareListParser

```

```

        {\noexpand\glsnumlistparser}{\delimN}}}%
    \@gls@dodedeflistparser
\fi
\let\makenoidxglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
    \let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
\ifglsautomake
    \renewcommand*{\@gls@doautomake}{%
        \protected@edef\gls@tmp{\@gls@automake@types}%
        \@for\@gls@type:=\gls@tmp\do{%
            \ifdefempty{\@gls@type}{%
                {\@gls@automake{\@gls@type}}%
            }%
        }%
    }%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}
\newcommand\@makeglossaries@warn@noprntglossary{%
    \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@noprntglossary
}
\newcommand*\makenoidxglossaries{%
    \def\glsindexingsetting{noidx}%
    \@domakeglossaries
    {%
        \renewcommand{\@gls@noref@warn}[1]{%
            \GlossariesWarning{Empty glossary for
                \string\printnoidxglossary[type={##1}].
                Rerun may be required (or you may have forgotten to use
                commands like \string\gls)}%
        }%
        \let\@gls@checkmkidxchars\@gobble
        \glsesclocationsfalse
        \let\@do@wrglossary\gls@noidxglossary
    }%
}

```

```

\let@gls@getgrouptitle@gls@noidx@getgrouptitle
\let@gls@checkseeallowed@relax
\renewcommand{\@do@seeglossary}[2]{%
  \protected@edef@gls@label{\glsdetoklabel{##1}}%
  \protected@write@auxout{}{%
    \string@gls@reference
      {\csname glo@%gls@label @type\endcsname}%
      {\@gls@label}%
      {%
        \string@glsseeformat##2}%
      }%
  }%
}%
\AtBeginDocument
{%
  \write@auxout{\string\providecommand\string@gls@reference[3]{}}%
}%
\def\warn@noprintglossary{%
  \GlossariesWarningNoLine{No \string\printnoidxglossary\space
    or \string\printnoidxglossaries ^^J
    found. (Remove \string\makenoidxglossaries\space if you
    don't want any glossaries.)^^JThis document will not have a glossary}%
}%
\let\warn@nomakeglossaries@relax
\let\makeglossaries@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
  \let@glo@assign@sortkey@glo@assign@sortkey
  \def@glo@sorttype@glo@default@sorttype}%
}%
\renewcommand*\new@glossaryentry[2]{%
  \PackageError{glossaries}{Glossary entries must be
    defined in the preamble^^Jwhen you use
    \string\makenoidxglossaries}%
  {Either move your definitions to the preamble or use
  \string\makeglossaries}%
}%
\renewcommand*{\glsentrynumberlist}[1]{%
  \letcs{@gls@loclist}{glo@glsdetoklabel{##1}@loclist}%
  \ifdef@gls@loclist
  {%
    \glsnoidxloclist{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for '##1'. Either
        a rerun is required or you haven't referenced the entry}%
    }%
  }%
}%

```

```

\renewcommand*{\glsdisplaynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \def\@gls@noidxloclist@sep{%
      \def\@gls@noidxloclist@sep{%
        \def\@gls@noidxloclist@sep{%
          \glsnumlistsep
        }%
      }%
    }%
    \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
  }%
  \def\@gls@noidxloclist@finalsep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
  \@gls@noidxloclist@finalsep
  \@gls@noidxloclist@prev
}%
{%
  ??\glsdoifexists{##1}%
  {%
    \GlossariesWarning{Missing location list for ‘##1’. Either
      a rerun is required or you haven’t referenced the entry}%
  }%
}%
\renewcommand*{\glsnumberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo@glsdetoklabel{##1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
  \let\@gls@org@glsseeformat\glsseeformat
  \let\glsnoidxdisplayloc##2\relax
  \let\glsseeformat##3\relax
  \ifdef\@gls@loclist
  {%
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
  \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
  \let\glsseeformat\@gls@org@glsseeformat
}%
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
\@glo@check@sortallowed\makenoidxglossaries

```



```

\let\@gls@savenonumberlist\gls@noidx@nonumberlist
}%
}
\@onlypreamble{\makenoidxglossaries}
\newcommand*\glsnumberlistloop}[3]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
\newcommand*\glsnoidxnumberlistloophandler}[1]{%
  #1%
}
\newcommand*\@no@makeglossaries}{%
  \PackageError{glossaries}{You can't use both
  \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
  together.}%
}
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
    is required to make \string\printnoidxglossary[type=#1] work}%
}
\newcommand*\gls@noidxglossary}{%
  \protected@write\@auxout{\gls@wrglossdisablelocationcmds}{%
    \string\@gls@reference
    {\csname glo@\@gls@label @type\endcsname}%
    {\@gls@label}%
    {\string\glsnoidxdisplayloc
     {\@glo@counterprefix}%
     {\@gls@counter}%
     {\@glsnumberformat}%
     {\@gls@locref}}%
    }%
  }%
}
\AtEndDocument{%
  \gls@writefiles
}
\newcommand*\@gls@writefiles}{%
  \foralllglossaries{\@glo@type}{%
    \ifcsundef{glo@\@glo@type @filetok}%
    {%
      \def\gls@tmp{}%
    }%
    {%
      \edef\gls@tmp{\expandafter\the
        \csname glo@\@glo@type @filetok\endcsname}%
    }%
    \ifx\gls@tmp\@empty
      \ifx\@glo@type\gls@defaulttype
        \GlossariesWarningNoLine{Glossary '\@glo@type' has no

```

```

        entries.^^JRemember to use package option ‘nomain’ if
you
        don’t want to^^Juse the main glossary}%
    \else
        \GlossariesWarningNoLine{Glossary ‘\@glo@type’ has no
            entries}%
    \fi
    \else
        \@glsopenfile{\glswrite}{\@glo@type}%
        \immediate\write\glswrite{%
            \expandafter\the
                \csname glo@\@glo@type @filetok\endcsname}%
        \immediate\closeout\glswrite
    \fi
}%
}
\newcommand*{\gls@glossary}[1]{%
    \@gls@glossary{#1}%
}
\newcommand*{\@gls@glossary}[2]{%
    \if@gls@debug
        \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
    \fi
}
\newcommand{\@gls@renewglossary}{%
    \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
    \let\@gls@renewglossary\@empty
}
\newcommand*{\gls@wrglossary}[2]{%
    \ifglssavewrites
        \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
        \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
            \expandafter{\@gls@tmp^^J}%
    \else
        \ifcsdef{glo@#1@file}%
            {%
                \expandafter\protected@write\csname glo@#1@file\endcsname{%
                    \gls@disablepagerefexpansion}{#2}%
            }%
            {%
                \ifignoredglossary{#1}{}%
                    {%
                        \GlossariesWarning{No file defined for glossary ‘#1’}%
                    }%
                }%
            }%
        \fi
    \endgroup\@esphack
}
\newcommand*{\@do@wrglossary}[1]{%
    \gls@ifnotmeasuring{\glswriteentry{#1}{\@do@wrglossary{#1}}}%
}

```

```

}
\newcommand*\glswriteentry}[2]{%
  \ifglsindexonlyfirst
    \ifglsused{#1}{#2}%
  \else
    #2%
  \fi
}
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
\newcommand*\gls@disablepagerefexpansion{%
\glswrglossdisablelocationcmds
\ifglsesclocations
  \@for\@gls@this:=\gls@protected@pagefmts\do
  {%
    \expandafter\let\@gls@this\relax
  }%
\fi
}
\newcommand{\glswrglossdisablelocationcmds}{}
\newcommand*\gls@alphpage}{\@Alph\c@page}
\newcommand*\gls@Alphpage}{\@Alph\c@page}
\newcommand*\gls@numberpage}{\number\c@page}
\newcommand*\gls@arabicpage}{\@arabic\c@page}
\newcommand*\gls@romanpage}{\romannumeral\c@page}
\newcommand*\gls@Romanpage}{\@Roman\c@page}
\newcommand*\glsaddprotectedpagefmt}[1]{%
  \ifglsesclocations
    \protected@eappto\gls@protected@pagefmts{\expandonce{\csname gls#1page\endcsname}}%
    \csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
    \protected@eappto\@wrglossarynumberhook{%
      \noexpand\let\expandonce{\csname org@gls#1\endcsname}%
      \expandonce{\csname#1\endcsname}%
      \noexpand\def\expandonce{\csname#1\endcsname}{%
        \noexpand\@wrglossary@pageformat
        \expandonce{\csname gls#1page\endcsname}%
        \expandonce{\csname org@gls#1\endcsname}%
      }%
    }%
  \else
    \GlsWarnAddProtectedPageFmt
  \fi
}
\newcommand{\GlsWarnAddProtectedPageFmt}{%
  \GlossariesWarning{\string\glsaddprotectedpagefmt\space has no
  effect with esclocations=false}%
}
\newcommand*\@wrglossarynumberhook{}
\newcommand{\@wrglossary@pageformat}[3]{%
  \ifx#3\c@page #1\else #2#3\fi
}

```

```

}
\newcommand*{\@do@wrglossary}[1]{%
  \ifgllesclocations
    \@do@esc@wrglossary{#1}%
  \else
    \@do@noesc@wrglossary{#1}%
  \fi
}
\newcommand{\glswrglossdisableanchorcmds}{\let\glstexorpdfstring\@secondoftwo}
\ifdef\texorpdfstring
  {\appto\glswrglossdisableanchorcmds{\let\texorpdfstring\@secondoftwo}}{}
\ifdef\pdfstringdefPreHook
  {\appto\glswrglossdisableanchorcmds{\pdfstringdefPreHook}}{}
\newcommand{\glswrglosslocationtarget}[1]{#1}
\newcommand{\glswrglosslocationtextfmt}[1]{#1}
\newcommand*{\@do@noesc@wrglossary}[1]{%
  \expandafter\def\expandafter\@glslocref\expandafter{\theglslentrycounter}%
  \expandafter\def\expandafter\@glsHlocref\expandafter{\theHglslentrycounter}%
  \ifx\@glsHlocref\@glslocref
    \def\@glo@counterprefix{}%
  \else
    \xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
  \fi
  \protected@edef\@gls@label{\glsdetoklabel{#1}}%
  \@do@@wrglossary
}
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodsfalse
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
  \begingroup
    \let\gls@orgthe\the
    \let\gls@orgnumber\number
    \let\gls@orgarabic\@arabic
    \let\gls@orgromannumeral\romannumeral
    \let\gls@orgalph\@alph
    \let\gls@orgAlph\@Alph
    \let\gls@orgRoman\@Roman
    \ifglswrallowprimitivemods
      \def\gls@the##1{%
        \ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
      \def\the{\expandafter\gls@the}%
      \def\gls@number##1{%
        \ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
      \def\number{\expandafter\gls@number}%
      \def\romannumeral##1{%
        \ifx##1\c@page \gls@romanpage \else \gls@orgromannumeral##1\fi}%
    \fi
    \def\@arabic##1{%
      \ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
    \def\@Roman##1{%

```

```

        \ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
    \ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
    \ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\gls@locref{\theglsentrycounter}%
\endgroup
\@gls@checkmkidxchars\@gls@locref
\expandafter\ifx\theglsentrycounter\theglsentrycounter\relax
\def\@glo@counterprefix{%
\else
\protected@edef\@gls@Hlocref{\theglsentrycounter}%
\@gls@checkmkidxchars\@gls@Hlocref
\@xp@gls@getcounterprefix{\@gls@locref}{\@gls@Hlocref}%
\fi
\protected@edef\@gls@label{\gls@detoklabel{#1}}%
\@do@@wrglossary
}
\newcommand*{\@do@@wrglossary}{%
\ifglxindy
\expandafter\@glo@check@mkidxrangechar\@gls@numberformat\@nil
\def\@glo@range{%
\expandafter\if\@glo@prefix\relax
\def\@glo@range{:open-range}%
\else
\expandafter\if\@glo@prefix\relax
\def\@glo@range{:close-range}%
\fi
\fi
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
(indexentry :tkey (\csname glo@\@gls@label @index\endcsname)
:locref \string"\@glo@counterprefix}{\@gls@locref}\string" %
:attr \string"\@gls@counter\@glo@suffix\string"
\@glo@range
)
}%
\else
\@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@gls@numberformat}%
{\@glo@counterprefix}%
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
\string\glossaryentry{\csname glo@\@gls@label @index\endcsname
\@gls@encapchar\@glo@numfmt}{\@gls@locref}}%
\fi
}
\newcommand*\@xp@gls@getcounterprefix[2]{%
\bgroup
\gls@wrglossdisableanchorcmds
\protected@edef\@do@gls@getcounterprefix{%

```

```

        \noexpand\egroup
        \noexpand\@gls@getcounterprefix{#1}{#2}%
    }%
    \do@gls@getcounterprefix
}
\newcommand*\@gls@getcounterprefix[2]{%
    \def\@gls@thisloc{#1}\def\@gls@thisHloc{#2}%
    \ifx\@gls@thisloc\@gls@thisHloc
        \def\@glo@counterprefix{}%
    \else
        \def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
            \def\@glo@tmp{##2}%
            \ifx\@glo@tmp\@empty
                \def\@glo@counterprefix{}%
            \else
                \def\@glo@counterprefix{##1}%
            \fi
        }%
        \@gls@get@counterprefix#2.#1\end@getprefix
        \ifx\@glo@counterprefix\@empty
            \GlossariesWarning{Hyper target ‘#2’ can’t be formed by
                prefixing^^Jlocation ‘#1’. You need to modify the
                definition of \string\theH\@gls@counter^^Jotherwise you
                will get the warning: "‘name{\@gls@counter.#1}’ has been^^J
                referenced but does not exist"%
        \fi
    \fi
}
\newcommand{\@do@seeglossary}[2]{%
    \def\@gls@xref{#2}%
    \@onelevel@sanitize\@gls@xref
    \@gls@checkmkidxchars\@gls@xref
    \ifglsxindy
        \gls@glossary{\csname glo@#1@type\endcsname}{%
            (indexentry
                :key (\csname glo@#1@index\endcsname)
                :xref (\string"\@gls@xref\string")
                :attr \string"see\string"
            )
        }%
    \else
        \gls@glossary{\csname glo@#1@type\endcsname}{%
            \string\glossaryentry{\csname glo@#1@index\endcsname
                \@gls@encapchar glsseeformat\@gls@xref}{Z}}%
        \fi
    }
}
\def\@gls@fixbraces#1#2#3\@nil{%
    \if#2[\relax
        \@gls@fixbraces#1#2#3\@end@fixbraces
    \else

```

```

        \def#1{#2#3}%
    \fi
}
\def\@gls@fixbraces#1[#2]#3\end@fixbraces{%
    \def#1{[#2]{#3}}%
}
\newrobustcmd*\glssee[3][\seename]{%
    \@do@seeglossary{#2}{#1}{#3}}
\newcommand*\@glssee[3][\seename]{%
    \glssee[#1]{#3}{#2}}
\newrobustcmd*\glsseeformat[3][\seename]{%
    \emph{#1} \glsseelist{#2}}
\newrobustcmd*\glsseelist[1]{%
    \let\@gls@dolast\relax
    \let\@gls@donext\relax
    \@for\@gls@thislabel:=#1\do{%
        \ifx\@xfor@nextelement\@nnil
            \@gls@dolast
        \else
            \@gls@donext
        \fi
        \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
        \let\@gls@dolast\glsseelastsep
        \let\@gls@donext\glsseesep
    }%
}
\newcommand*\glsseelastsep{\space\andname\space}
\newcommand*\glsseesep{, }
\newrobustcmd*\glsseeitem[1]{\gls hyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*\glsseeitemformat[1]{%
    \ifgls hasshort{#1}{\glsentrytext{#1}}{\glsentryname{#1}}%
}
\newcommand*\gls@save@numberlist[1]{%
    \ifgls@savenumberlist
        \toks@{#1}%
        \protected@edef\@do@writeaux@info{%
            \noexpand\csgdef{glo@glscurrententrylabel @numberlist}{\the\toks@}%
        }%
        \@onelevel@sanitize\@do@writeaux@info
        \protected@write\@auxout{\@do@writeaux@info}%
    \fi
}
\newcommand*\warn@noprntglossary{}%
\ifcsundef{printglossary}{}%
{%
    \@gls@warnonglossdefined
    \undef\printglossary
}
\newcommand*\@printgloss@checkexists@noignored[2]{%
    \ifglossaryexists{#1}%

```

```

{#2}%
{%
  \ifignoredglossary{#1}%
  {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
  {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}%
}
\newcommand*{\@printgloss@checkexists@allowignored}[2]{%
  \s@ifglossaryexists{#1}%
  {#2}%
  {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\newcommand*{\printglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@glossary}%
}
\newcommand*{\printglossaries}{%
  \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}
\newcommand*{\printnoidxglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@noidx@glossary}%
}
\newcommand*{\printnoidxglossaries}{%
  \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
\newcommand*{\@printgloss@setsort}{}
\newcommand*{\@gls@preglossaryhook}{}
\newcommand{\@printglossary}[2]{%
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{\csname @glo@type\@glo@type @title\endcsname}%
  \def\glossarytoctitle{\glossarytitle}%
  \let\org@glossarytitle\glossarytitle
  \def\@glossarystyle{%
    \ifx\@glossary@default@style\relax
      \GlossariesWarning{No default glossary style provided \MessageBreak
        for the glossary ‘\@glo@type’. \MessageBreak
        Using fallback. \MessageBreak
        To fix this set the style with \MessageBreak
        \string\setglossarystyle\space or use the \MessageBreak
        style key=value option}%
    \fi
  }%
  \def\gls@dotoc@title{\glssettoctitle{\@glo@type}}%
  \let\@org@glossaryentrynumbers\glossaryentrynumbers
  \bgroup
  \@printgloss@setsort
  \setkeys{printgloss}{#1}%
  \@printgloss@checkexists{\@glo@type}%
}

```



```

{%
\ifx\glossarytitle\org@glossarytitle
\else
\expandafter\let\csname @glotype@\@glo@type @title\endcsname
\glossarytitle
\fi
\let\currentglossary\@glo@type
\let\org@glossaryentrynumbers\glossaryentrynumbers
\let\glsnonextpages\@glsnonextpages
\let\glsnextpages\@glsnextpages
\let\nopostdesc\@nopostdesc
\gls@dotoc@title
\@glossarystyle
\let\gls@org@glossaryentryfield\glossentry
\let\gls@org@glossarysubentryfield\subglossentry
\renewcommand{\glossentry}[1]{%
\protected@xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
\gls@org@glossaryentryfield{##1}%
}%
\renewcommand{\subglossentry}[2]{%
\protected@xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
\gls@org@glossarysubentryfield{##1}{##2}%
}%
\@gls@preglossaryhook
#2%
}%
\egroup
\global\let\glossaryentrynumbers\@org@glossaryentrynumbers
\global\let\warn@noprintglossary\relax
}
\newcommand{\@print@glossary}{%
\makeatletter
\@input@{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
\IfFileExists{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
{}%
{\null}%
\ifglsxindy
\gls@xdy@set@lang{\@glo@type}%
\edef\@do@auxoutstuff{%
\noexpand\AtEndDocument{%
\noexpand\immediate\noexpand\write\@auxout{%
\string\providecommand\string\@xdylanguage[2]{}}%
\noexpand\immediate\noexpand\write\@auxout{%
\string\@xdylanguage{\@glo@type}{\@gls@langmod}}%
}%
}%
\@do@auxoutstuff
\edef\@do@auxoutstuff{%
\noexpand\AtEndDocument{%
\noexpand\immediate\noexpand\write\@auxout{%

```

```

        \string\providecommand\string\@gls@codepage[2]{}%
        \noexpand\immediate\noexpand\write\@auxout{%
        \string\@gls@codepage{\@glo@type}{\gls@codepage}}%
    }%
}%
\@do@auxoutstuff
\fi
\renewcommand*{\@warn@nomakeglossaries}{%
    \GlossariesWarningNoLine{\string\makeglossaries\space
    hasn't been used,^^Jthe glossaries will not be updated}%
}%
}
\newcommand*{\@glo@sortentries}[2]{%
    \glosortentrieswarning
    \def\@glo@sortinglist{}%
    \def\@glo@sortinghandler{#1}%
    \protected@edef\@glo@type{#2}%
    \forlistcsloop{\@glo@do@sortentries}{\glsref@#2}%
    \csdef{\glsref@#2}{%
    \@for\@this@label:=\@glo@sortinglist\do{%
        \xifinlistcs{\@this@label}{\glsref@#2}%
        {%
            \listcsxadd{\glsref@#2}{\@this@label}%
        }%
        \ifcsdef{\@glo@sortingchildren@\@this@label}%
        {%
            \@glo@addchildren{#2}{\@this@label}%
        }%
        }%
    }%
}
}
\newcommand*{\@glo@addchildren}[2]{%
    \bgroup
    \letcs{\@glo@childlist}{\@glo@sortingchildren@#2}%
    \@for\@this@childlabel:=\@glo@childlist\do
    {%
        \xifinlistcs{\@this@childlabel}{\glsref@#1}%
        {%
            \listcsxadd{\glsref@#1}{\@this@childlabel}%
        }%
        \ifcsdef{\@glo@sortingchildren@\@this@childlabel}%
        {%
            \@glo@addchildren{#1}{\@this@childlabel}%
        }%
        }%
    }%
    \egroup
}

```

```

}
\newcommand*{\@glo@do@sortentries}[1]{%
  \ifglshasparent{#1}%
  {%
    \protected@edef\@glo@parent{\csuse{glo@\glsdetoklabel{#1}@parent}}%
    \ifcsundef{glo@sortingchildren@\@glo@parent}%
    {%
      \csdef{glo@sortingchildren@\@glo@parent}{}%
    }%
    {}%
    \expandafter\@glo@sortedinsert
    \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
    \xifinlistcs{\@glo@parent}{\glsref{\@glo@type}}%
    {%
    }%
    {%
      \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
    }%
  }%
  {}%
  \@glo@sortedinsert{\@glo@sortinglist}{#1}%
}%
}
\newcommand*{\@glo@sortedinsert}[2]{%
  \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@\glsorder}%
        {%
            \@glo@sortentries{\csuse{@glo@sorthandler@\glsorder}}{#1}%
        }%
        {%
            \PackageError{glossaries}{Unknown sort handler ‘\glsorder’}{-%
        }%
    }%
    {%
}

```

```

        \PackageError{glossaries}{Conflicting sort options:^^J
        \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
        \string\printnoidxglossary[sort=standard]}{}}%
    }%
}
\newcommand*\@glo@sortmacro@case}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
        \@glo@sortentries{\@glo@sorthandler@case}{#1}%
    }%
    {%
        \PackageError{glossaries}{Conflicting sort options:^^J
        \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
        \string\printnoidxglossary[sort=case]}{}}%
    }%
}
\newcommand*\@glo@sortmacro@nocase}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
        \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
    }%
    {%
        \PackageError{glossaries}{Conflicting sort options:^^J
        \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
        \string\printnoidxglossary[sort=nocase]}{}}%
    }%
}
\newcommand*\@glo@sortmacro@def}[1]{%
    \def\@glo@sortinglist{}%
    \forlslentries[#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}%
{%
\glsnoidxprenumberlist{#1}%
\glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
}%
}%
}%
}%
{%
\letcs{\@gls@sort}{glo\glsdetoklabel{#1}@sort}%
\expandafter\glo@grabfirst\@gls@sort{}{}\@nil
\ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
{}%
{%
\ifdefempty{\@gls@currentlettergroup}{}%
{%
\global\let\@glo@thislettergrp\@glo@thislettergrp
\glsgroupskip
}%
\glsgroupheading{\@glo@thislettergrp}%
}%
\global\let\@gls@currentlettergroup\@glo@thislettergrp
\ifdefvoid{\@gls@loclist}
{%
\glossentry{#1}{}%
}%
{%
\glossentry{#1}%
{%
\glsnoidxprenumberlist{#1}%
\glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
}%
}%
}%

```

```

    }%
  }
  \newcommand*\glsnoidxloclist}[1]{%
    \def\@gls@noidxloclist@sep{}%
    \def\@gls@noidxloclist@prev{}%
    \forlistloop{\glsnoidxloclisthandler}{#1}%
  }
  \newcommand*\glsnoidxloclisthandler}[1]{%
    \ifdefstring{\@gls@noidxloclist@prev}{#1}%
    {%
    }%
    {%
      \@gls@noidxloclist@sep
      #1%
      \def\@gls@noidxloclist@sep{\delimN}%
      \def\@gls@noidxloclist@prev{#1}%
    }%
  }
  \newcommand*\glsnoidxdisplayloclisthandler}[1]{%
    \ifdefstring{\@gls@noidxloclist@prev}{#1}%
    {%
    }%
    {%
      \@gls@noidxloclist@sep
      \@gls@noidxloclist@prev
      \def\@gls@noidxloclist@prev{#1}%
    }%
  }
  \newcommand*\glsnoidxdisplayloc[4]{%
    \setentrycounter[#1]{#2}%
    \csuse{#3}{#4}%
  }
  \newcommand*\@gls@reference}[3]{%
    \glsdoifexistsorwarn{#2}%
    {%
      \ifcsundef{\@glsref@#1}{\csgdef{\@glsref@#1}{}}{%
        \ifinlistcs{#2}{\@glsref@#1}%
        }{%
          \listcsgadd{\@glsref@#1}{#2}%
          \ifcsundef{glo@\glsdetoklabel{#2}@loclist}%
          {\csgdef{glo@\glsdetoklabel{#2}@loclist}{}}%
          }{%
            \listcsgadd{glo@\glsdetoklabel{#2}@loclist}{#3}%
          }%
    }%
  }
  \define@key{printgloss}{type}{\def\@glo@type{#1}}
  \define@key{printgloss}{title}{%
    \def\glossarytitle{#1}%
    \let\gls@dotocitle\relax
  }

```



```

\define@key{printgloss}{toctitle}{%
\def\glossarytoctitle{#1}%
\let\gls@dottoctitle\relax
}
\define@key{printgloss}{style}{%
\ifcsundef{@glsstyle@#1}%
{%
\PackageError{glossaries}%
{Glossary style ‘#1’ undefined}{}%
}%
}%
\def\@glossarystyle{\setglossarystyle{#1}}%
}%
}
\define@choicekey{printgloss}{numberedsection}%
[\gls@numberedsection@val\gls@numberedsection@nr]%
{false,nolabel,autolabel,nameref}[nolabel]%
{%
\ifcase\gls@numberedsection@nr\relax
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{}%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{}%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{\label{\glsautoprefix\@glo@type}}%
\or
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}}%
\fi
}
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
\csuse{glsnogroupskip#1}%
}
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
\csuse{glsnopostdot#1}%
}
}
\newcommand*\GlsEntryCounterLabelPrefix{glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
\csuse{glsentrycounter#1}%
\@gls@define@glossaryentrycounter
}
}
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
\csuse{glsesubentrycounter#1}%
\@gls@define@glossarysubentrycounter
}
}
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%

```

```

\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\currentglsstepsubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglsstepsubentry}%
  \fi

```

```

}
\newcommand*{\glsrefentry}[1]{%
\ifglentrycounter
\ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
\else
\ifglssubentrycounter
\ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
\else
\gls{#1}%
\fi
\fi
}
\newcommand*{\glentrycounterlabel}{%
\ifglentrycounter
\theglossaryentry.\space
\fi
}
\newcommand*{\glssubentrycounterlabel}{%
\ifglssubentrycounter
\theglossarysubentry)\space
\fi
}
\newcommand*{\glentryitem}[1]{%
\ifglentrycounter
\glsstepentry{#1}\glentrycounterlabel
\else
\glsresetsubentrycounter
\fi
}
\newcommand*{\glssubentryitem}[1]{%
\ifglssubentrycounter
\glsstepsubentry{#1}\glssubentrycounterlabel
\fi
}
\ifcsundef{theglossary}%
{%
\newenvironment{theglossary}{-}{-}%
}%
{%
\@gls@warnontheGLOSSdefined
\renewenvironment{theglossary}{-}{-}%
}
\newcommand*{\glossaryheader}{}
\newcommand*{\glstarget}[2]{\@glstarget{\glo@linkprefix#1}{#2}}
\newcommand*{\glossentry}[2]{\glossentryname{#1} #2}
\newcommand*{\glossentryname}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
\expandafter\glsnamefont\expandafter{\glo@name}%
}
}

```

```

    }%
  }
  \newcommand*{\Glossentryname}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \glsnamefont{\Glsentryname{#1}}%
    }%
  }
  \glsmfuaddmap{\glossentryname}{\Glossentryname}
  \newcommand*{\glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \glsentrydesc{#1}%
    }%
  }
  \newcommand*{\Glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \Glsentrydesc{#1}%
    }%
  }
  \glsmfuaddmap{\glossentrydesc}{\Glossentrydesc}
  \newcommand*{\glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \glsentrysymbol{#1}%
    }%
  }
  \newcommand*{\Glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \Glsentrysymbol{#1}%
    }%
  }
  \glsmfuaddmap{\glossentrysymbol}{\Glossentrysymbol}
  \newcommand{\subglossentry}[3]{\glossentryname{#2} #3}
  \newcommand*{\glsgroupskip}{=}
  \newcommand*{\glsgroupheading}[1]{=}
  \newrobustcmd*{\glsgetgrouptitle}[1]{%
    \@gls@getgrouptitle{#1}{\@gls@grptitle}%
    \@gls@grptitle
  }
  \ExplSyntaxOn
  \newcommand*{\@gls@getgrouptitle}[2]{%
    \def#2{#1}%
    \regex_match:nnT { ~\A(.|glsnumbers|glssymbols)\Z } { #1 }
    { \ifcsundef{#1groupname}{ }\letcs#2{#1groupname} }
  }
  \ExplSyntaxOff
  \newcommand*{\@gls@noidx@getgrouptitle}[2]{%

```

```

\DTLifint{#1}%
{\edef#2{\char#1\relax}}%
{%
  \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
}%
}
\ExplSyntaxOn
\newcommand*{\glsgetgrouplabel}[1]{%
  \str_if_eq:eeTF {#1} { \glssymbolsgroupname }
  { \glssymbols }
  {
    \str_if_eq:eeTF {#1} { \glsnumbersgroupname }
    { \glsnumbers }
    { #1 }
  }
}
\ExplSyntaxOff
\newcommand*{\setentrycounter}[2][ ]{%
  \def\@glo@counterprefix{#1}%
  \ifx\@glo@counterprefix\empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1.}%
  \fi
  \def\glsentrycounter{#2}%
}
\newcommand*{\setglossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \csname @glsstyle@#1\endcsname
  }%
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\newcommand{\newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
\newcommand{\renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%

```

```

    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
\newcommand*{\glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\glsnumber#1{#1}%
}%
{%
  \def\glsnumber#1{\xp@glsnumber{#1}}
}
\newrobustcmd{\xp@glsnumber}[1]{%
  \glsnumber#1\nohyperpage{ }\@nil
}
\def\@glsnumber#1\nohyperpage#2#3\@nil{%
  \ifx\#1\%
  \else
    \gls@delimR#1\delimR\delimR\%
  \fi
  \ifx\#2\%
  \else
    #2%
  \fi
  \ifx\#3\%
  \else
    \glsnumber#3\@nil
  \fi
}
\def\gls@delimR#1\delimR #2\delimR #3\%
\ifx\#2\%
  \gls@delimN{#1}%
\else
  \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
\fi
}
\def\gls@delimN#1{\gls@delimN#1\delimN \delimN\}
\def\gls@delimN#1\delimN #2\delimN#3\%
\ifx\#3\%
  \@gls@numberlink{#1}%
\else
  \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
\fi
}
\def\@gls@numberlink#1{%
  \begingroup
  \toks@={}%
  \@gls@removespaces#1 \@nil

```

```

\endgroup
}
\def\@gls@removespaces#1 #2\@nil{%
\toks@=\expandafter{\the\toks@#1}%
\ifx\#2\%
\edef\@gls@x{\the\toks@}%
\ifx\@gls@x\empty
\else
\bgroup
\glswrglossdisableanchorcmds
\protected@edef\@gls@tmp{\glsentrycounter\@glo@counterprefix
\glswrglosslocationtarget{\@gls@x}}%
\@onelevel@sanitize\@gls@tmp
\edef\@gls@x{\noexpand\egroup
\noexpand\hyperlink{\@gls@tmp}%
{\noexpand\glswrglosslocationtextfmt{\the\toks@}}%
}%
\@gls@x
\fi
\else
\@gls@ReturnAfterFi{%
\@gls@removespaces#2\@nil
}%
\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
\newcommand*\hyperrm}[1]{\textrm{\glsnumber{#1}}}
\newcommand*\hypersf}[1]{\textsf{\glsnumber{#1}}}
\newcommand*\hypertt}[1]{\texttt{\glsnumber{#1}}}
\newcommand*\hyperbf}[1]{\textbf{\glsnumber{#1}}}
\newcommand*\hypermd}[1]{\textmd{\glsnumber{#1}}}
\newcommand*\hyperit}[1]{\textit{\glsnumber{#1}}}
\newcommand*\hypersl}[1]{\textsl{\glsnumber{#1}}}
\newcommand*\hyperup}[1]{\textup{\glsnumber{#1}}}
\newcommand*\hypersc}[1]{\textsc{\glsnumber{#1}}}
\newcommand*\hyperemph}[1]{\emph{\glsnumber{#1}}}
\newcommand{\oldacronym}[4][\@gls@label]{%
\def\@gls@label{#2}%
\newacronym[#4]{#1}{#2}{#3}%
\ifcsundef{xspace}%
{%
\expandafter\edef\csname#1\endcsname{%
\noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}%
}%
}%
}%
\expandafter\edef\csname#1\endcsname{%
\noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
\noexpand\gls{#1}\noexpand\xspace}%
}%
}

```

```

}%
}
\newcommand{\newacronym}[4][\%
\gls@newacronym{#1}{#2}{#3}{#4}%
}
\newcommand{\gls@newacronym}[4]{%
\renewcommand{\gls@newacronym}[4]{%
\PackageError{glossaries}{incompatible acronym style:
rollback may be required}%
{something has gone wrong with \string\newacronym. Try
\string\usepackage{glossaries}[=v4.49]}%
}%
\setacronymstyle{long-short}%
\newacronym[#1]{#2}{#3}{#4}%
}
\newcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}
\newrobustcmd*{\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
\newcommand*{\glsshortkey}{short}
\newcommand*{\glsshortpluralkey}{shortplural}
\newcommand*{\glslongkey}{long}
\newcommand*{\glslongpluralkey}{longplural}
\newcommand{\acrlinkfullformat}[5]{%
\acrfullformat{#1}{#3}{#4}[#5]{#2}{#3}{#4}[]}%
}
\newcommand{\acrfullformat}[2]{#1\glspace{#2}}
\newrobustcmd*{\glsspace}{\space}
\newcommand*{\acrfullfmt}[3]{%
\acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newcommand*{\Acrfullfmt}[3]{%
\acrlinkfullformat{\@Acrlong}{\@Acrshort}{#1}{#2}{#3}%
}
\newcommand*{\ACRfullfmt}[3]{%
\acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
\newcommand*{\acrfullplfmt}[3]{%
\acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newcommand*{\Acrfullplfmt}[3]{%
\acrlinkfullformat{\@Acrlongpl}{\@Acrshortpl}{#1}{#2}{#3}%
}
\newcommand*{\ACRfullplfmt}[3]{%
\acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfull}{\@gls@hyp@opt\ns@acrfull}
\newcommand*{ns@acrfull}[2][\%
\new@ifnextchar[{\@acrfull{#1}{#2}}%
{\@acrfull{#1}{#2}[]}%
}
\def\@acrfull#1#2[#3]{%

```



```

\acrfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\Acrfull}{\@gls@hyp@opt\ns@Acrfull}
\glsmfuaddmap{\acrfull}{\Acrfull}
\newcommand*\ns@Acrfull[2][{}]{%
  \new@ifnextchar[{\@Acrfull{#1}{#2}}%
    {\@Acrfull{#1}{#2}[]}%
}
\def\@Acrfull#1#2[#3]{%
  \Acrfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfull}{\@gls@hyp@opt\ns@ACRfull}
\newcommand*\ns@ACRfull[2][{}]{%
  \new@ifnextchar[{\@ACRfull{#1}{#2}}%
    {\@ACRfull{#1}{#2}[]}%
}
\def\@ACRfull#1#2[#3]{%
  \ACRfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfullpl}{\@gls@hyp@opt\ns@acrfullpl}
\newcommand*\ns@acrfullpl[2][{}]{%
  \new@ifnextchar[{\@acrfullpl{#1}{#2}}%
    {\@acrfullpl{#1}{#2}[]}%
}
\def\@acrfullpl#1#2[#3]{%
  \acrfullplfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\Acrfullpl}{\@gls@hyp@opt\ns@Acrfullpl}
\glsmfuaddmap{\acrfullpl}{\Acrfullpl}
\newcommand*\ns@Acrfullpl[2][{}]{%
  \new@ifnextchar[{\@Acrfullpl{#1}{#2}}%
    {\@Acrfullpl{#1}{#2}[]}%
}
\def\@Acrfullpl#1#2[#3]{%
  \Acrfullplfmt{#1}{#2}{#3}%
}
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand{\acrnameformat}[2]{\acronymfont{#1}}
\newtoks\glskeylisttok
\newtoks\glslabeltok
\newtoks\glsshorttok

```

```

\newtoks\glslongtok
\newcommand*{\newacronymhook}{}
\newcommand*{\SetGenericNewAcronym}{%
  \let\@Gls@entryname\@Gls@acentryname
  \renewcommand{\newacronym}[4][]{%
    \ifdefempty{\@glsacronymlists}%
    {%
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
    }%
  }%
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \glslongtok{##4}%
  \newacronymhook
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\expandonce{\acronymentry{##2}}},%
      sort={\acronymsort{\the\glsshorttok}{\the\glslongtok}},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\glspluralsuffix},%
      \GenericAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsuppercase{\genacrfullformat{##2}{##3}}}%
  }%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsuppercase{\genplacrfullformat{##2}{##3}}}%
  }%
\renewcommand*{\glsentryfull}[1]{\genacrfullformat{##1}{}}%
\renewcommand*{\Glsentryfull}[1]{\Genacrfullformat{##1}{}}%

```

```

\renewcommand*{\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}%
\renewcommand*{\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}%
}
\newcommand*{\GenericAcronymFields}{description={\the\glslongtok}}
\newcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
\newcommand*{\acronymsort}[2]{##1}
\newcommand*{\setacronymstyle}[1]{%
\ifcsundef{@glsacr@dispstyle@##1}
{%
\PackageError{glossaries}{Undefined acronym style ‘##1’}{}%
}%
{%
\ifdefempty{\@glsacronymlists}%
{%
\DeclareAcronymList{\acronymtype}%
}%
{}%
\SetGenericNewAcronym
\GlsUseAcrStyleDefs{##1}%
\@for\@gls@type:=\@glsacronymlists\do{%
\defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{##1}}%
}%
}%
}
\newcommand*{\newacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@##1}%
{%
\PackageError{glossaries}{Acronym style ‘##1’ already exists}{}%
}%
{%
\csdef{@glsacr@dispstyle@##1}{##2}%
\csdef{@glsacr@styledefs@##1}{##3}%
}%
}
\newcommand*{\renewacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@##1}%
{%
\csdef{@glsacr@dispstyle@##1}{##2}%
\csdef{@glsacr@styledefs@##1}{##3}%
}%
{%
\PackageError{glossaries}{Acronym style ‘##1’ doesn’t exist}{}%
}%
}
\newcommand*{\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@##1}}
\newcommand*{\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@##1}}
\newacronymstyle{long-short}%
{%
\ifglsashaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%

```

```

{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
\ifglsashaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
}

```

```

\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newcommand*\glsacspace}[1]{%
\settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{##1}})}%
\ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
\ifglsahaslong{\glslabel}{\glsacspace}{\glsacspace}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshort{##1}}##2\space
(\glsentrylong{##1})%
}%
\renewcommand*\Genacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshort{##1}}##2\space
(\glsentrylong{##1})%
}%
\renewcommand*\genplacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
(\glsentrylongpl{##1})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
(\glsentrylongpl{##1})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{%
\GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*\acronymfont}[1]{\textsc{##1}}%
\renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
}
\newacronymstyle{long-sm-short}%
{%
\GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*\acronymfont}[1]{\textsmaller{##1}}%
}

```

```

    \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
  }
  \newacronymstyle{sc-short-long}%
  {%
    \GlsUseAcrEntryDispStyle{short-long}%
  }%
  {%
    \GlsUseAcrStyleDefs{short-long}%
    \renewcommand{\acronymfont}[1]{\textsc{##1}}%
    \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
  }
  \newacronymstyle{sm-short-long}%
  {%
    \GlsUseAcrEntryDispStyle{short-long}%
  }%
  {%
    \GlsUseAcrStyleDefs{short-long}%
    \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
    \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
  }
  \newacronymstyle{long-short-desc}%
  {%
    \GlsUseAcrEntryDispStyle{long-short}%
  }%
  {%
    \GlsUseAcrStyleDefs{long-short}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymsort}[2]{##2}%
    \renewcommand*{\acronymentry}[1]{%
      \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})%
    }
  }
  \newacronymstyle{long-sp-short-desc}%
  {%
    \GlsUseAcrEntryDispStyle{long-sp-short}%
  }%
  {%
    \GlsUseAcrStyleDefs{long-sp-short}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymsort}[2]{##2}%
    \renewcommand*{\acronymentry}[1]{%
      \glsentrylong{##1}\glsacspace{##1}(\acronymfont{\glsentryshort{##1}})%
    }
  }
  \newacronymstyle{long-sc-short-desc}%
  {%
    \GlsUseAcrEntryDispStyle{long-sc-short}%
  }%
  {%
    \GlsUseAcrStyleDefs{long-sc-short}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymsort}[2]{##2}%
  }

```

```

\renewcommand*\acronymentry}[1]{%
  \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext

```

```

{%
  \ifglshaslong{\glslabel}%
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \glstrylongpl{\glslabel}\glsinsert
      }%
      {%
        \Glsentrylongpl{\glslabel}\glsinsert
      }%
      {%
        \glsupercase
        {\glstrylongpl{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glscapscase
    {%
      \glstrylong{\glslabel}\glsinsert
    }%
    {%
      \Glsentrylong{\glslabel}\glsinsert
    }%
    {%
      \glsupercase
      {\glstrylong{\glslabel}\glsinsert}%
    }%
  }%
  }%
  {%
    \glsgenentryfmt
  }%
  }%
  {\glscustomtext\glsinsert}%
}%
{%
  \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
  \renewcommand*\acrfullfmt}[3]{%
    \glslink[##1]{##2}{\glstrylong{##2}##3\space
      (\acronymfont{\glstryshort{##2}})}%
  \renewcommand*\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{\Glsentrylong{##2}##3\space
      (\acronymfont{\glstryshort{##2}})}%
  \renewcommand*\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
      \glsupercase{\glstrylong{##2}##3\space
        (\acronymfont{\glstryshort{##2}})}%
    }%
  \renewcommand*\acrfullplfmt}[3]{%

```



```

\glslink[##1]{##2}{\glsentrylongpl{##2}##3\space
(\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{\Glsentrylongpl{##2}##3\space
(\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%
\glsupercase{\glsentrylongpl{##2}##3\space
(\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\glsentryfull}[1]{%
\glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
\Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
\glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
\Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
\GlsUseAcrEntryDispStyle{dua}%
}%
{%
\GlsUseAcrStyleDefs{dua}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymentry}[1]{\glsentrylong{##1}}%
\renewcommand*{\acronymsort}[2]{##2}%
}%
\newacronymstyle{footnote}%
{%
\ifglsashaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshort{##1}}##2%
\protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\firstacronymfont{\Glsentryshort{##1}}##2%
\protect\footnote{\Glsentrylong{##1}}%
}

```

```

}%
\renewcommand*\genplacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\acrpluralsuffix{\glsacrpluralsuffix}%
\renewcommand*\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
  \glsupercase{\acronymfont{\glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}}}%
\renewcommand*\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
  \glsupercase{\acronymfont{\glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}}}%
\renewcommand*\glsentryfull}[1]{%
  \acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*\Glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*\glsentryfullpl}[1]{%
  \acronymfont{\glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
\renewcommand*\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
}
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymfont}[1]{\textsc{##1}}%

```

```

\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}%
\newacronymstyle{footnote-sm}%
{%
\GlsUseAcrEntryDispStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}%
\newacronymstyle{footnote-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{footnote-sc-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
\GlsUseAcrStyleDefs{footnote-sc}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{footnote-sm-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote-sm}%
}%
{%
\GlsUseAcrStyleDefs{footnote-sm}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newcommand*{\DefineAcronymSynonyms}{%
\let\acs\acrshort
\let\Acs\Acrshort
\let\acsp\acrshortpl
\let\Acsp\Acrshortpl

```

```

\let\acl\acrlong
\let\aclp\acrlongpl
\let\Acl\Acrlong
\let\Aclp\Acrlongpl
\let\acf\acrfull
\let\acfp\acrfullpl
\let\Acf\Acrfull
\let\Acfp\Acrfullpl
\let\ac\gls
\let\Ac\Gls
\let\acp\glspl
\let\Acp\Glspl
}
\ifglsacrshortcuts
  \DefineAcronymSynonyms
\fi
\newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
\newcommand*{\acrlinkfootnote}[3]{%
  \footnote{\glslink{#1}{#2}{#3}}%
}
\newcommand*{\acrnoflinkfootnote}[3]{%
  \footnote{#3}%
}
\newrobustcmd*{\glsdoparenifnotempty}[2]{%
  \protected@edef\gls@tmp{#1}%
  \ifdefempty\gls@tmp
  {}%
  {%
    \ifx\gls@tmp\@gls@default@value
    \else
    \space (#2{#1})%
    \fi
  }%
}
\newcommand*{\@gls@deprecated@acr@warning}[3]{%
  \GlossariesWarning{Acronym style option #1 deprecated.
  Use \string\setacronymstyle{#2} with just glossaries.sty or
  use \string\setabbreviationstyle[acronym]{#3} with
  glossaries-extra.sty}%
}
\RequirePackage{glossary-hypernav}[=v4.52]
\@gls@loadlist
\@gls@loadlong
\@gls@loadsuper
\@gls@loadtree
\ifx\@glossary@default@style\relax
\else

```

```

\setglossarystyle{\@glossary@default@style}
\fi
\newcommand*{\showgloparent}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
}
\newcommand*{\showglolevel}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
}
\newcommand*{\showglotext}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
}
\newcommand*{\showgloplural}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@plural\endcsname
}
\newcommand*{\showglofirst}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@first\endcsname
}
\newcommand*{\showglofirstpl}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@firstpl\endcsname
}
\newcommand*{\showglotype}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@type\endcsname
}
\newcommand*{\showglocounter}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@counter\endcsname
}
\newcommand*{\showglouseri}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@useri\endcsname
}
\newcommand*{\showglouserii}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@userii\endcsname
}
\newcommand*{\showglouseriii}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@useriii\endcsname
}
\newcommand*{\showglouseriv}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@useriv\endcsname
}
\newcommand*{\showglouserv}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@userv\endcsname
}
\newcommand*{\showglouservi}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@uservi\endcsname
}
\newcommand*{\showglongame}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@name\endcsname
}
\newcommand*{\showglodesc}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@desc\endcsname
}
}

```

```

\newcommand*{\showglodescplural}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@descplural\endcsname
}
\newcommand*{\showglosort}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@sort\endcsname
}
\newcommand*{\showglosymbol}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@symbol\endcsname
}
\newcommand*{\showglosymbolplural}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@symbolplural\endcsname
}
\newcommand*{\showgloshort}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@short\endcsname
}
\newcommand*{\showglolong}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@long\endcsname
}
\newcommand*{\showgloindex}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@index\endcsname
}
\newcommand*{\showgloflag}[1]{%
  \expandafter\show\csname ifglo@glstdetoklabel{#1}@flag\endcsname
}
\newcommand*{\showgloclist}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@loclist\endcsname
}
\newcommand*{\showglofield}[2]{%
  \csshow{glo@glstdetoklabel{#1}@#2}%
}
\newcommand*{\showacronymlists}{%
  \show\@glsacronymlists
}
\newcommand*{\showglossaries}{%
  \show\@glo@types
}
\newcommand*{\showglossaryin}[1]{%
  \expandafter\show\csname @glotype@#1@in\endcsname
}
\newcommand*{\showglossaryout}[1]{%
  \expandafter\show\csname @glotype@#1@out\endcsname
}
\newcommand*{\showglossarytitle}[1]{%
  \expandafter\show\csname @glotype@#1@title\endcsname
}
\newcommand*{\showglossarycounter}[1]{%
  \expandafter\show\csname @glotype@#1@counter\endcsname
}
\newcommand*{\showglossaryentries}[1]{%
  \expandafter\show\csname glolist@#1\endcsname
}

```

```
}
```

8.36 Rollback v4.52 (glossary-hypernav-2022-11-03.sty)

Version 4.52 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossary-hypernav}[2022/11/03 v4.52 (NLCT)]
\newcommand*\glsnavhyperlink}[3][\currentglossary]{%
  \protected@edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%
  \glslink{\glsnavhyperlinkname{#1}{#2}}{#3}%
}
\newcommand*\glsnavhyperlinkname}[2]{\glsn:#1@#2}
\newcommand*\glsnavhypertarget}{\protect\@gls@navhypertarget}
\newcommand*\@gls@navhypertarget}[3][\currentglossary]{%
  \glsnavhypertarget{#1}{#2}{#3}%
}
\newcommand*\@gls@navhypertarget}[3]{%
  \protected@write\@auxout{}{\string\@gls@hypergroup{#1}{#2}}%
  \gls@target{\glsnavhyperlinkname{#1}{#2}}{#3}%
  \expandafter\let
    \expandafter\@gls@list\csname @gls@hypergroup@list@#1\endcsname
  \@for\@gls@elem:=\@gls@list\do{%
    \ifthenelse{equal{\@gls@elem}{#2}}{\@endfortrue}{}%
  }
  \if@endfor
  \else
    \GlossariesWarningNoLine{Navigation panel
      for glossary type ‘#1’^^Jmissing group ‘#2’}%
    \gdef\gls@hypergroup@rprerun{%
      \GlossariesWarningNoLine{Navigation panel
        has changed. Rerun LaTeX}}%
  \fi
}
\let\gls@hypergroup@rprerun\relax
\AtEndDocument{\gls@hypergroup@rprerun}
\newcommand*\@gls@hypergroup}[2]{%
  \ifcsdef{\@gls@hypergroup@list@#1}%
  {\protected@csxappto{\@gls@hypergroup@list@#1}{, #2}}%
  {\protected@csxdef{\@gls@hypergroup@list@#1}{#2}}%
}
\newcommand*\glsnavigation}{%
  \def\@gls@between{}%
  \ifcsundef{\@gls@hypergroup@list@\currentglossary}%
  {%
    \def\@gls@list{}%
  }%
  {%
    \letcs\@gls@list{\@gls@hypergroup@list@\currentglossary}%
  }%
  \@for\@gls@tmp:=\@gls@list\do{%
    \@gls@between
  }
}
```

```

    \@gls@getgrouptitle{\@gls@tmp}{\@gls@grptitle}%
    \expandafter\glsnavhyperlink\expandafter{\@gls@tmp}{\@gls@grptitle}%
    \let\@gls@between\glshypernavsep
  }%
}
\newcommand*\glshypernavsep{\space\textbar\space}
\newcommand*\glssymbolnav{%
  \glsnavhyperlink{glssymbols}{\glsgetgrouptitle{glssymbols}}%
  \glshypernavsep
  \glsnavhyperlink{glsnumbers}{\glsgetgrouptitle{glsnumbers}}%
  \glshypernavsep
}

```


Change History

1.01 – 2007-05-17	
General: Added range facility in format key	126
\writeist: Added spaces after \delimN and \delimR in ist file	179
1.04 – 2007-08-03	
General: Added \glstextformat	113
1.05 – 2007-08-10	
\glossarysection: added	
\@mkboth to	
\glossarysection	51
\gls@defglossaryentry:	
Changed the default value of the sort key to just the value of the name key	95
1.07 – 2007-09-13	
\@gls@link: fixed bug caused by \theglsentrycounter setting the page number too soon	123
\glsadd: fixed bug caused by \theglsentrycounter setting the page number too soon	176
1.08 – 2007-10-13	
General: Added babel support	45
altlistgroup: changed altlistgroup style to use \glsgetgrouptitle	278
listgroup: changed listgroup style to use \glsgetgrouptitle	277
1.1 – 2008-02-22	
General: amsgen now loaded (\new@ifnextchar needed)	6
\@glossarysection: numbered sections and auto label added	52
\@gls@tmpb: changed \toksdef to \newtoks	128
\@gls@toc: numberline added	54
\@p@glossarysection: numbered sections and auto label added	53
numberedsection:	
numberedsection package option added	10
numberline: numberline option added	10
\setglossarysection: new	52
translate: translate option added	30
1.12 – 2008-03-08	
General: added check for \hypertarget separate to \hyperlink (memoir defines \hyperlink but not \hypertarget)	135
\@GLSpl: now uses	
\glsentrydescplural and \glsentrysymbolplural instead of \glsentrydesc and \glsentrysymbol	140
\@GLspl0: now uses	
\glsentrydescplural and \glsentrysymbolplural instead of \glsentrydesc and \glsentrysymbol	139
\@glspl0: now uses	
\glsentrydescplural and \glsentrysymbolplural instead of \glsentrydesc and \glsentrysymbol	138
descriptionplural: new	75
\gls@defglossaryentry:	
Changed default first plural to be first key with s appended (was text key with s appended)	95
descriptionplural support added	95
symbolplural support added	95
\Glsentrydescplural: New	168
\glsentrydescplural: New	168
\Glsentrysymbolplural: New	169
\glsentrysymbolplural: New	169
symbolplural: new	77
1.13 – 2008-05-10	
General: fixed bug that ignored 3rd parameter	142–150
\ACRfullpl: new	244
\Acrfullpl: new	244
\acrfullpl: new	244
\acrpluralsuffix: New	241
\gls@defglossaryentry:	
Changed default first value	95

Changed default firstplural value	95	checking if footnote option has been used	139
Removed restriction on only using <code>\newglossaryentry</code> in the preamble	100	<code>\@gls@</code> : Test glossary type is <code>\acronymtype</code> in addition to checking if footnote option has been used	136
<code>\newacronym</code> : Removed restriction on only using <code>\newacronym</code> in the preamble	240	<code>\@glsdisp</code> : Test glossary type is <code>\acronymtype</code> in addition to checking if footnote option has been used	141
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	138
General: added nonnumberlist key to <code>\printglossary</code>	228	<code>\@gls@defglossaryentry</code> : Changed def to let	95
added numberedsection key to <code>\printglossary</code>	226	1.17 – 2008-12-26	
<code>\@gls@hypergroup</code> : new	272	General: added xindy support	32
<code>\firstacronymfont</code> : new	245	<code>\@do@esc@wrglossary</code> : new	205
<code>\glsautoprefix</code> : new	10	<code>\@do@seeglossary</code> : new	209
<code>\glsnavhyperlink</code> : changed <code>\edef</code> to <code>\protected@edef</code>	270	<code>\@glo@storeentry</code> : new	102
<code>\glsnavhypertarget</code> : added write to aux file	271	<code>\@gls@glossary</code> : changed definition to use <code>\index</code> instead of <code>\@index</code>	201
<code>\glsnavigation</code> : changed to only use labels for groups that are present	273	<code>\@glsdefaultplural</code> : new	80
1.15 – 2008-08-15		<code>\@glsdefaultsort</code> : new	81
<code>\@gls@link</code> : added <code>\glslabel</code>	123	<code>\@gls@hypernumber</code> : new	237
<code>\@gls@defglossaryentry</code> : check for <code>\@glo@first</code> in description	99	<code>\@gls@noname</code> : new	80
check for <code>\@glo@text</code> in symbol	99	<code>\@gls@nonextpages</code> : new	228
<code>\@gls@hypergroup@rerun</code> : new	272	<code>\@gls@defglossaryentry</code> : added nonnumberlist key	96
<code>\glsnavhypertarget</code> : added check if rerun required	271	added parent key	95
<code>\@gls@settoctitle</code> : new	44	added see key	95
<code>\printglossary</code> : changed the way the TOC title is set	212	Stored main part of entry format when entry is defined	100
1.16 – 2008-08-27		<code>\@gls@suffixF</code> : new	49
<code>\@GLS@</code> : Test glossary type is <code>\acronymtype</code> in addition to checking if footnote option has been used	138	<code>\@gls@suffixFF</code> : new	49
<code>\@GLSpl</code> : Test glossary type is <code>\acronymtype</code> in addition to checking if footnote option has been used	140	<code>\@gls@wrglossary</code> : modified to allow for xindy support	202
<code>\@Gls@</code> : Test glossary type is <code>\acronymtype</code> in addition to checking if footnote option has been used	137	<code>\@gls@hyperlink</code> : new	175
<code>\@Glspl@</code> : Test glossary type is <code>\acronymtype</code> in addition to		<code>\@gls@hypernumber</code> : modified to allow material to be attached to location	237
		<code>\@glsnavhyperlink</code> : replaced <code>\hyperlink</code> to <code>\@glslink</code>	270

<code>\glsnavhypertarget</code> : replaced <code>\hypertarget</code> to <code>\@gls@target</code>	271	<code>\@printglossary</code> : changed <code>\xdef</code> to <code>\protected\xdef</code>	215
<code>\glssee</code> : new	210	<code>\gls@defglossaryentry</code> : changed <code>\xdef</code> to <code>\protected\xdef</code>	97, 100
<code>\glsseeformat</code> : new	210	<code>\glsaddprotectedpagefmt</code> : changed <code>\eappto</code> to <code>\protected@eappto</code>	203
<code>\glsSetSuffixF</code> : new	49	2.01 – 2009 May 30	
<code>\glsSetSuffixFF</code> : new	49	<code>\@gls@link</code> : moved <code>\@do@wrglossary</code> before term is displayed to prevent unwanted whatsit	124
<code>\ifglxindy</code> : new	32	<code>\forallglossaries</code> : replaced <code>\ifthenelse</code> with <code>\ifx</code>	63
<code>\istfilename</code> : added xindy support	48	<code>\forglentries</code> : replaced <code>\ifthenelse</code> with <code>\ifx</code>	64
<code>\newglossarystyle</code> : made <code>\newglossarystyle long</code>	236	<code>\glsdefmain</code> : new	19
<code>nonumberlist</code> : new	78	<code>\glsdescwidth</code> : changed <code>\linewidth</code> to <code>\hsize</code>	280, 303
<code>\nopostdesc</code> : new	47	<code>\glslistdottedwidth</code> : changed <code>\linewidth</code> to <code>\hsize</code>	279
<code>order</code> : order package option added	32	<code>\glspagelistwidth</code> : changed <code>\linewidth</code> to <code>\hsize</code>	280, 303
<code>parent</code> : new	78	<code>nomain</code> : added <code>nomain</code> package option	20
<code>\printglossary</code> : added check to determine if <code>\printglossary</code> is already defined	212	<code>\writeist</code> : removed <code>item_02</code> - no such <code>makeindex</code> key	184
added print language to aux file	212	2.02 – 2007-07-13	
<code>see</code> : new	77	<code>\@printglossary</code> : suppressed warning globally rather than locally	215
<code>\writeist</code> : added xindy support	179	2.02 – 2009-07-13	
1.18 – 2009-01-14		<code>\glossarysection</code> : changed <code>\@mkboth</code> to <code>\glossarymark</code>	51
<code>\@gls@loadlist</code> : new	13	<code>\gls@defglossaryentry</code> : Changed default value of <code>sort</code> to <code>\@gls@defaultsort</code>	95
<code>\@gls@loadlong</code> : new	12	moved sort sanitization to <code>\newglossaryentry</code>	99
<code>\@gls@loadsuper</code> : new	12	<code>\gls@target</code> : new	231
<code>\@gls@loadtree</code> : new	13	<code>nolist</code> : new	13
<code>\gls@defglossaryentry</code> : Changed default value of <code>sort</code> to <code>\@gls@defaultsort</code>	95	<code>nolong</code> : new	12
moved sort sanitization to <code>\newglossaryentry</code>	99	<code>nostyles</code> : new	13
<code>\gls@target</code> : new	231	<code>nosuper</code> : new	13
<code>nolist</code> : new	13	<code>notree</code> : new	13
<code>nolong</code> : new	12	<code>\oldacronym</code> : new	240
<code>nostyles</code> : new	13	<code>sort</code> : moved sanitization to <code>\newglossaryentry</code>	76
<code>nosuper</code> : new	13	1.19 – 2009-03-02	
<code>notree</code> : new	13	<code>\gls@clearpage</code> : new	54
<code>\oldacronym</code> : new	240	<code>\glsdisp</code> : new	140
<code>sort</code> : moved sanitization to <code>\newglossaryentry</code>	76	1.46 – 2021-09-20	
1.19 – 2009-03-02		General: changed <code>\edef</code> to <code>\protected@edef</code>	371
<code>\gls@clearpage</code> : new	54		
<code>\glsdisp</code> : new	140		
1.46 – 2021-09-20			
General: changed <code>\edef</code> to <code>\protected@edef</code>	371		

Moved entry existence check to avoid duplicate code	123	\glsadd: fixed bug that ignored counter	176
\@glsdisp: Added check for hyperfirst	141	\Glsentryuseri: new	171
\@glspl@: Added check for hyperfirst	138	\glsentryuseri: new	171
\gls glossarymark: Added check to see if it's already defined	51	\Glsentryuserii: new	171
hyperfirst: new	31	\glsentryuserii: new	171
2.04 – 2009-11-10		\Glsentryuseriii: new	171
\@GLS@: Changed test to check if glossary type has been identified as a list of acronyms	138	\glsentryuseriii: new	171
\@GLSp1: Changed test to check if glossary type has been identified as a list of acronyms	140	\Glsentryuseriv: new	172
\@Gls@: Changed test to check if glossary type has been identified as a list of acronyms	137	\glsentryuseriv: new	172
\@GLSp1@: Changed test to check if glossary type has been identified as a list of acronyms	139	\Glsentryuserv: new	172
\@glossaryentryfield: new	101	\glsentryuserv: new	172
\@glossarysubentryfield: new	101	\Glsentryuservi: new	172
\@gls@: Changed test to check if glossary type has been identified as a list of acronyms	136	\glsentryuservi: new	172
\@glsacronymlists: new	21	\ns@newglossary: added check to determine if \gls@<type>@display and \gls@<type>@displayfirst have been defined.	73
\@glsdisp: Changed test to check if glossary type has been identified as a list of acronyms	141	\SetAcronymLists: new	22
\@glspl@: Changed test to check if glossary type has been identified as a list of acronyms	138	\SetDefaultAcronymDisplayStyle: new	260
\@newglossaryentryposthook: new	101	\SetDefaultAcronymStyle: new	260
\@newglossaryentryprehook: new	101	\SetDescriptionAcronymDisplayStyle: new	261
acronymlists: new	22	\SetDescriptionDUAAcronymDisplayStyle: new	261
\DeclareAcronymList: new	21	\SetDescriptionFootnoteAcronymDisplayStyle: new	260
\DefineAcronymSynonyms: new	258	\SetDUADisplayStyle: new	262
\gls@defglossaryentry: added user1-6 keys	96	\SetFootnoteAcronymDisplayStyle: new	261
		\SetSmallAcronymDisplayStyle: new	262
		2.05 – 2010-02-06	
		\@glsdisp: Added closing brace. Patch provided by Sergiu Dotenco	141
		Removed spurious brace. Patch provided by Sergiu Dotenco	141
		\writeist: Added \string before opening and closing braces. Patch provided by Segiu Dotenco	184
		2.06 – 2010-06-14	
		\altnewglossary: new	73
		\CustomAcronymFields: new	263
		\CustomNewAcronymDef: new	263
		\SetCustomDisplayStyle: new	263
		\SetCustomStyle: new	263

2.07 – 2010-07-10		<code>\@xdyattributelist</code> : new	54
General: <code>glsadd</code> format key stored		<code>\acrfootnote</code> : new	260
in <code>\@glsnumberformat</code> (was		<code>\ACRfull</code> : added starred version	243
mistakenly stored in		<code>\Acrfull</code> : added starred version	243
<code>\@glo@format</code>)	176	<code>\acrfull</code> : added starred version	243
3.0 – 2010-07-12		<code>\ACRfullpl</code> : added starred	
<code>\@makeglossary</code> : Added check for		version	244
<code>savewrites</code>	189	<code>\Acrfullpl</code> : added starred	
<code>\@gls@wrglossary</code> : modified to		version	244
take into account <code>savewrites</code>	202	<code>\acrfullpl</code> : added starred	
3.0 – 2010/03/31		version	244
<code>\@set@glo@numformat</code> : added 4th		<code>\acrlinkfootnote</code> : new	260
argument	126	<code>\acrnolinkfootnote</code> : new	260
3.0 – 2011-04-02		compatible-2.07:	
General: added prefix to		compatible-2.07 option added	38
hyperlink	238	counter: replaced <code>\@ifundefined</code>	
etoolbox now loaded	6	with <code>\ifcsundef</code>	77
replaced <code>\@ifundefined</code> with		<code>counterwithin</code> : new	14
<code>\ifcsundef</code>	43, 46, 119, 226	<code>entrycounter</code> : new	14
<code>\@do@esc@wrglossary</code> : added		<code>glossaryentry</code> : new	14
check for hyper location		<code>\glossarysection</code> : replaced	
prefix	207	<code>\@ifundefined</code> with	
modified to use new format	205	<code>\ifcsundef</code>	51
<code>\@@glossarysec</code> : replaced		<code>\glossarystyle</code> : replaced	
<code>\@ifundefined</code> with		<code>\@ifundefined</code> with	
<code>\ifcsundef</code>	10	<code>\ifcsundef</code>	235
<code>\@do@seeglossary</code> : Sanitize and		<code>glossarysubentry</code> : new	15
escape cross-referencing		<code>\gls@codepage</code> : replaced	
information	209	<code>\@ifundefined</code> with	
<code>\@gls@counterwithin</code> : new	15	<code>\ifcsundef</code>	33
<code>\@gls@ifinlist</code> : new	55	<code>\gls@defglossaryentry</code> : added	
<code>\@gls@link</code> : added		<code>\@gls@defsort</code>	99
<code>\@gls@saveentrycounter</code>	124	added short and long keys	96
added <code>\@gls@setsort</code>	124	replaced <code>\@ifundefined</code> with	
<code>\@gls@saveentrycounter</code> : new	124	<code>\ifcsundef</code>	96
<code>\@gls@setupsort@def</code> : new	17	<code>\gls@doclearpage</code> : replaced	
<code>\@gls@setupsort@standard</code> : new	16	<code>\@ifundefined</code> with	
<code>\@gls@setupsort@use</code> : new	17	<code>\ifcsundef</code>	53
<code>\@gls@xdy@locationlist</code> : new	58	<code>\glsadd</code> : added	
<code>\@gls@link</code> : replaced		<code>\@gls@saveentrycounter</code>	176
<code>\@ifundefined</code> with		<code>\GlsAddXdyCounters</code> : new	55
<code>\ifcsundef</code>	134	<code>\glsentrycounterlabel</code> : new	230
<code>\@glsnextpages</code> : new	228	<code>\glsentryitem</code> : new	230
<code>\@print@glossary</code> : replaced		<code>\Glsentrylong</code> : new	173
<code>\@ifundefined</code> with		<code>\glsentrylong</code> : new	173
<code>\ifcsundef</code>	215	<code>\Glsentrylongpl</code> : new	173
<code>\@printglossary</code> : added		<code>\glsentrylongpl</code> : new	173
<code>\currentglossary</code>	214	<code>\Glsentryshort</code> : new	173
added <code>\glsnextpages</code>	214	<code>\glsentryshort</code> : new	172
make toctitle default to title	214	<code>\Glsentryshortpl</code> : new	173

<code>\glsentryshortpl</code> : new	173	<code>\oldacronym</code> : replaced	
<code>\glsgetgrouptitle</code> : replaced		<code>\@ifundefined</code> with	
<code>\ifcsundef</code>	234	<code>\ifcsundef</code>	240
<code>\glsglossarymark</code> : replaced		<code>\printglossary</code> : replaced	
<code>\@ifundefined</code> with		<code>\@ifundefined</code> with	
<code>\ifcsundef</code>	51	<code>\ifcsundef</code>	212
<code>\glshyperlink</code> : changed default		<code>savewrites</code> : new	38
from <code>\glsentryname</code> to		<code>see</code> : added	
<code>\glsentrytext</code>	175	<code>\@glo@seeautonumberlist</code>	77
<code>\glshypernumber</code> : replaced		<code>seeautonumberlist</code> : new	12
<code>\@ifundefined</code> with		<code>\SetDescriptionFootnoteAcronymDisplayStyle</code> :	
<code>\ifcsundef</code>	237	expanded options link	
<code>\glsnumberformat</code> : replaced		options	260
<code>\@ifundefined</code> with		<code>\setentrycounter</code> : added	
<code>\ifcsundef</code>	49	optional argument	235
<code>\glsrefentry</code> : new	229	<code>short</code> : new	80
<code>\glsresetsubentrycounter</code> :		<code>shortplural</code> : new	80
new	229	<code>\showacronymlists</code> : new	268
<code>\glsseeitem</code> : hyperlink uses		<code>\showglocounter</code> : new	265
<code>\glsseeitemformat</code> instead of		<code>\showglodesc</code> : new	266
<code>\glsentryname</code>	211	<code>\showglodescplural</code> : new	266
<code>\glsseeitemformat</code> : new	211	<code>\showglofirst</code> : new	264
<code>\glssortnumberfmt</code> : new	16	<code>\showglofirstpl</code> : new	264
<code>\glsstepentry</code> : new	229	<code>\showgloflag</code> : new	267
<code>\glsstepsubentry</code> : new	229	<code>\showgloindex</code> : new	267
<code>\glssubentrycounterlabel</code> :		<code>\showglolevel</code> : new	264
new	230	<code>\showglongame</code> : new	266
<code>\glssubentryitem</code> : new	230	<code>\showgloparent</code> : new	263
<code>\ifglossaryexists</code> : replaced		<code>\showgloplural</code> : new	264
<code>\@ifundefined</code> with		<code>\showglosort</code> : new	266
<code>\ifcsundef</code>	64	<code>\showglossaries</code> : new	268
<code>\ifglsentryexists</code> : replaced		<code>\showglossarycounter</code> : new	269
<code>\@ifundefined</code> with		<code>\showglossaryentries</code> : new	269
<code>\ifcsundef</code>	65	<code>\showglossaryin</code> : new	268
<code>\istfile</code> : deprecated	199	<code>\showglossaryout</code> : new	268
<code>long</code> : new	80	<code>\showglossarytitle</code> : new	268
<code>longplural</code> : new	80	<code>\showglosymbol</code> : new	266
<code>\newglossaryentry</code> : replaced		<code>\showglosymbolplural</code> : new	267
<code>\DeclareRobustCommand</code> with		<code>\showglotext</code> : new	264
<code>\newrobustcmd</code>	83	<code>\showglotype</code> : new	264
<code>\newglossarystyle</code> : replaced		<code>\showglouserii</code> : new	265
<code>\@ifundefined</code> with		<code>\showglouseriii</code> : new	265
<code>\ifcsundef</code>	236	<code>\showglouseriv</code> : new	265
<code>nonumberlist</code> : now boolean	78	<code>\showglouservi</code> : new	265
<code>\ns@newglossary</code> : added		<code>\showglouservi</code> : new	266
<code>\@gls@defsortcount</code>	73	<code>sort</code> : new	15
replaced <code>\@ifundefined</code> with		<code>subentrycounter</code> : new	15
<code>\ifcsundef</code>	73		

theglossary: replaced		\glsname: made robust	145
\@ifundefined with		\GLSpl: made robust	140
\ifcsundef	230	\Glspl: made robust	139
\writeist: added xindy-only		\glspl: made robust	138
macro definitions to glossary		\GLSplplural: made robust	144
open tag	182	\GLSsymbol: made robust	149
modified to support new		\Glsymbol: made robust	149
format	179	\glsymbol: made robust	148
3.01 – 2011-04-12		\GLSsymbolplural: made robust	150
General: made robust	137	\Glsymbolplural: made robust	150
\@glswritefiles: added check		\glsymbolplural: made robust	149
for empty glossaries	200	\GLstext: made robust	142
\ACRfull: made robust	243	\glstext: made robust	142
\Acrfull: made robust	243	\GLSuseri: made robust	151
\acrfull: made robust	243	\Glsuseri: made robust	151
\acrfullformat: removed		\glsuseri: made robust	150
\acronymfont as it should		\GLSuserii: made robust	152
already be set in the second		\Glsuserii: made robust	152
argument.	242	\glsuserii: made robust	151
\ACRfullpl: made robust	244	\GLSuseriii: made robust	153
\Acrfullpl: made robust	244	\Glsuseriii: made robust	153
\acrfullpl: made robust	244	\glsuseriii: made robust	152
\ACRlong: made robust	162	\GLSuseriv: made robust	154
\Acrlong: made robust	161	\Glsuseriv: made robust	154
\acrlong: made robust	160	\glsuseriv: made robust	153
\ACRlongpl: made robust	164	\GLSuseriv: made robust	155
\Acrlongpl: made robust	163	\Glsuseriv: made robust	155
\acrlongpl: made robust	162	\glsuseriv: made robust	154
\ACRshort: made robust	158	\GLSuservi: made robust	156
\Acrshort: made robust	157	\Glsuservi: made robust	156
\acrshort: made robust	156	\glsuservi: made robust	155
\ACRshortpl: made robust	160	3.02 – 2012-05-19	
\Acrshortpl: made robust	159	\glsnumlistlastsep: new	175
\acrshortpl: made robust	158	\glsnumlistsep: new	175
\GLs: made robust	136	3.02 – 2012-05-21	
\glsadd: made robust	176	\@do@wrglossary: changed	
\glsaddall: made robust	177	\@glslocref to	
\GLSdesc: made robust	147	\theglsentrycounter	208
\Glsdesc: made robust	147	\do@wrglossary: changed	
\glsdesc: made robust	146	\do@wr@glossary to test for	
\GLSdescplural: made robust	148	indexonlyfirst option; put old	
\Glsdescplural: made robust	148	\do@wr@glossary code into	
\glsdescplural: made robust	147	\@do@wrglossary	202
\glsfirst: made robust	143	\@gls@missingnumberlist: new	81
\GLSfirstplural: made robust	145	\@glswritefiles: added check	
\Glsfirstplural: made robust	145	for existence of token in case	
\glsfirstplural: made robust	144	\makeglossaries has been	
\glslink: made robust	122	omitted	200
\GLSname: made robust	146	\@printglossary: add a way to	
\Glsname: made robust	146	fetch current entry label	215

<code>\gls@defglossaryentry</code> : added numberlist element	99	<code>\glspostinline</code> : replaced “.” with <code>\glspostdescription</code>	275
<code>\gls@save@numberlist</code> : new . .	211	index: added check for glsnogroupskip	318
<code>\gls@wrglossary</code> : added check for glossary file defined . . .	202	list: added check for glsnogroupskip	277
<code>\glsdisplaynumberlist</code> : new .	174	long: added check for glsnogroupskip	281
<code>\glsentrycounter</code> : set default value	124	long3col: added check for glsnogroupskip	282
<code>\Glsentryfull</code> : fixed bug (replaced <code>\glsentryshortpl</code> with <code>\glsentryshort</code>)	174	long4col: added check for glsnogroupskip	284
<code>\glsentryfullpl</code> : fixed bug (replaced <code>\glsentryshort</code> with <code>\glsentryshortpl</code>) . .	174	longragged: added check for glsnogroupskip	292
<code>\glsentrynumberlist</code> : new . . .	174	longragged3col: added check for glsnogroupskip	294
<code>\glsmoveentry</code> : new	101	nogroupskip: new	14
<code>\glsresetentrycounter</code> : new .	229	nopostdot: new	13
<code>\ifglshaschildren</code> : new	67	super: added check for glsnogroupskip	303
<code>\ifglshasparent</code> : new	67	super3col: added check for glsnogroupskip	305
indexonlyfirst: new	31	super4col: added check for glsnogroupskip	307
<code>\makeglossaries</code> : added list parser	194	superragged: added check for glsnogroupskip	311
<code>\renewglossarystyle</code> : new . . .	236	superragged3col: added check for glsnogroupskip	313
savenumberlist: new	12	tree: added check for glsnogroupskip	320
<code>\showglossaryentries</code> : fixed misspelt command	269	treenoname: added check for glsnogroupskip	321
<code>\SmallNewAcronymDef</code> : fixed broken short and long plural	262		
ucmark: new	14		
3.03 – 2012/09/21			
General: allow title to set toctitle	226		
<code>\@gls@sanitizesort</code> : new	25	3.04 – 2012-11-11	
<code>\@gls@setupsort@standard</code> : used <code>\@gls@sanitizesort</code>	16	<code>altlist</code> : replaced <code>\newline</code> with paragraph break	278
<code>\@printglossary</code> : allow title to override default toctitle . . .	213	3.04 – 2012-11-18	
<code>altlongragged4col</code> : added check for glsnogroupskip	295	General: Added check for doc package	6
<code>altsuperragged4col</code> : added check for glsnogroupskip . .	314	added datatool-base as a required package	6
<code>almtree</code> : added check for glsnogroupskip	324	added local key	120
<code>\glsinlinedescformat</code> : new . .	275	<code>\@do@wrglossary</code> : changed <code>\theglsentrycounter</code> back to <code>\@glslocref</code>	208
<code>\glsinlineemptydescformat</code> : new	275	<code>\@do@esc@wrglossary</code> : modified to compensate for possible incorrect page number	206
<code>\glsinlinenameformat</code> : new . .	275	<code>\@gls@esc@sdq</code> : unsanitize <code>\gls@numberpage</code> , <code>\gls@alphpage</code> ,	
<code>\glsinlinepostchild</code> : new . . .	275		
<code>\glsinlinesubdescformat</code> : new	276		
<code>\glsinlinesubnameformat</code> : new	276		

\gls@Alphpage and			
\gls@romanpage	127		
\@print@glossary: Moved aux			
write to end of document to			
prevent unwanted whatsit			
occurring here.	215		
\gls@Alphpage: new	203		
\gls@alphpage: new	203		
\gls@disablepagerefexpansion:			
new	202		
\gls@numberpage: new	203		
\gls@protected@pagefmts: new	202		
\gls@romanpage: new	203		
\glsdefmain: added check for doc			
package	19		
\glsorg@endtheglossary: new	6		
\glsorg@theglossary: new	6		
\PrintChanges: new	6		
3.05 – 2013-04-21			
\@do@esc@wrglossary: add			
Roman case. Fixed bugs in			
the else statements	206		
\@gls@link: added check for			
“nohypertypes”	123		
\gls@protected@pagefmts:			
added Roman to list	202		
\gls@Romanpage: new	203		
\glsgetgrouplabel: fixed bug			
(typo in \equal)	234		
\glsmcols: new	297		
mcolalttree: replaced ‘2’ with			
\glsmcols	301		
mcolindex: replaced ‘2’ with			
\glsmcols	297		
mcolindexspannav: replaced ‘2’			
with \glsmcols	298		
mcoltree: replaced ‘2’ with			
\glsmcols	298		
mcoltreenoname: replaced ‘2’			
with \glsmcols	300		
mcoltreesspannav: replaced ‘2’			
with \glsmcols	299		
\nopostdesc: made robust	47		
3.05 – 2013/04/21			
\@gls@nohyperlist: new	22		
\GlsDeclareNoHyperList: new	22		
nohypertypes: new	23		
3.06 – 2013/06/17			
\@xdy@main@language: Changed			
back to using \language	33		
\findrootlanguage: Obsoleted	62		
3.07 – 2013-07-05			
\@gls@link: fixed bug that failed			
to find entry in list	123		
\glossarypreamble: modified to			
work with			
\setglossarypreamble	50		
\gls@docclearpage: added check			
for openright	53		
\glspostdescription: Added			
spacefactor code	13		
\GlsSetXdyCodePage: Added			
check for fontspec	62		
\SetDescriptionAcronymDisplayStyle:			
now using			
\glsdoparenifnotempty	261		
\setglossarypreamble: new	50		
3.08a – 2013-08-30			
altlist: updated altlist style to			
use \glossentry and			
\subglossentry	278		
inline: updated inline style to			
use \glossentry and			
\subglossentry	274		
list: updated list style to use			
\glossentry and			
\subglossentry	277		
listdotted: updated listdotted			
style to use \glossentry and			
\subglossentry	279		
3.08a – 2013-09-28			
General: added nogroupskip key			
to \printglossary	227		
removed definition of			
\@glossaryentryfield	360		
removed definition of			
\@glossarysubentryfield	360		
\@glo@storeentry: no longer			
need to check for special			
characters in any of the fields			
other than sort	102		
updated for \glossentry	102		
\@glossaryentryfield: switched			
to \glossentry	101		
\@glossarysubentryfield:			
switched to \subglossentry	101		
\@gls@sanitizedesc: deprecated	24		
altlongragged4col: updated to			
use \glossentry and			
\subglossentry	295		

alttree: updated to use		superragged: updated to use	
\glossentry and		\glossentry and	
\subglossentry	323	\subglossentry	310
\compatibleglossentry: new ..	231	tree: updated to use	
\compatiblesubglossentry:		\glossentry and	
new	232	\subglossentry	319
\glossaryentryfield:		3.09a – 2013-10-09	
deprecated	232	\@gls@assign@symbolplural@field:	
\glossarystyle: deprecated ..	235	new	25
\glossentry: new	231	\@gls@default@value: new ...	76
\Glossentrydesc: new	232	\Glsentrydesc: made robust ..	168
\glossentrydesc: new	232	\Glsentrydescplural: made	
\Glossentryname: new	231	robust	168
\glossentryname: new	231	\Glsentryfirst: made robust ..	169
\Glossentrysymbol: new	232	\Glsentryfirstplural: made	
\glossentrysymbol: new	232	robust	170
\gls@assign@desc@field: new ..	24	\Glsentryfull: made robust ..	174
\gls@assign@descplural@field:		\Glsentryfullpl: made robust	174
new	25	\Glsentrylong: made robust ..	173
\gls@assign@field: new	83	\Glsentrylongpl: made robust	173
\gls@ifnotmeasuring: new ...	104	\Glsentryname: made robust ..	166
\glsaddallunused: new	177	\Glsentryplural: made robust	169
\glsexpandfields: new	83	\Glsentryshort: made robust ..	173
\glsnoexpandfields: new	83	\Glsentryshorttpl: made robust	173
\glssee: made robust	210	\Glsentrysymbol: made robust	169
\glsseeformat: made robust ..	210	\Glsentrysymbolplural: made	
\glsseeitem: made robust	211	robust	169
\glsseeitem: made robust	210	\Glsentrytext: made robust ..	168
\ifglsdescsuppressed: new ..	68	\Glsentryuseri: made robust ..	171
\ifglshasdesc: new	67	\Glsentryuserii: made robust	171
\ifglshassymbol: new	68	\Glsentryuseriii: made robust	171
index: added paragraph break at		\Glsentryuseriv: made robust	172
end of environment	317	\Glsentryuserv: made robust ..	172
updated to use \glossentry		\Glsentryuservi: made robust	172
and \subglossentry	318	\glstextup: new	241
long: updated to use		\ifglshassymbol: changed test	
\glossentry and		to check for	
\subglossentry	281	\@gls@default@symbol	68
longragged: updated to use		3.10a – 2013-09-28	
\glossentry and		\gls@assign@type@field: new ..	24
\subglossentry	292	3.10a – 2013-10-13	
longragged3col: updated to use		\@gls@keymap: new	86
\glossentry and		\@gls@provide@newglossary:	
\subglossentry	293	new	71
\setglossarystyle: new	235	\@gls@writedef: new	86
\setglossentrycompatibility:		\@glsdefaultplural: Obsolete ..	80
new	232	\@glsnodesc: new	80
\subglossentry: new	233	\@print@glossary: Added	
		providecommand code to aux	
		file	216

<code>\gls@defglossaryentry:</code>		changed to just use	
Changed to using		<code>\glsentrytext</code>	142
<code>\@gls@default@value</code>	95	changed to just use	
new	95	<code>\Glsentryuseriii</code>	153
<code>\gls@writedefhook:</code> new	94	changed to just use	
<code>\makeglossaries:</code> Added		<code>\glsentryuseriii</code>	153
providecommand code to aux		changed to just use	
file	193	<code>\Glsentryuserii</code>	152
<code>\new@glossaryentry:</code> new	84	changed to just use	
<code>\ns@newglossary:</code> added		<code>\glsentryuserii</code>	152
<code>\@gls@provide@newglossary</code>	73	changed to just use	
3.11a – 2013-10-15		<code>\Glsentryuseriv</code>	154
General: added <code>\glslabel</code> ,		changed to just use	
<code>\glsifplural</code> , <code>\glscapscase</code> ,		<code>\glsentryuseriv</code>	154
<code>\glsinsert</code> and		changed to just use	
<code>\glscustomtext</code>	157–164	<code>\Glsentryuseri</code>	151
changed to just use		changed to just use	
<code>\Glsentrydescplural</code>	148	<code>\glsentryuseri</code>	151
changed to just use		changed to just use	
<code>\glsentrydescplural</code>	148	<code>\Glsentryuservi</code>	156
changed to just use		changed to just use	
<code>\Glsentrydesc</code>	147	<code>\glsentryuservi</code>	156
changed to just use		changed to just use	
<code>\glsentrydesc</code>	147	<code>\Glsentryuserv</code>	155
changed to just use		changed to just use	
<code>\Glsentryfirstplural</code>	145	<code>\glsentryuserv</code>	155
changed to just use		Now requires <code>textcase</code>	5
<code>\glsentryfirstplural</code>	145	<code>\@ACRlong:</code> added <code>\glslabel</code> ,	
changed to just use		<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\Glsentryfirst</code>	143	<code>\glsinsert</code> and	
changed to just use		<code>\glscustomtext</code>	355
<code>\glsentryfirst</code>	143	<code>\@ACRshort:</code> added <code>\glslabel</code> ,	
changed to just use		<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\Glsentryname</code>	146	<code>\glsinsert</code> and	
changed to just use		<code>\glscustomtext</code>	354
<code>\glsentryname</code>	146	<code>\@Acrlong:</code> added <code>\glslabel</code> ,	
changed to just use		<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\Glsentryplural</code>	144	<code>\glsinsert</code> and	
changed to just use		<code>\glscustomtext</code>	355
<code>\glsentryplural</code>	144	<code>\@Acrshort:</code> added <code>\glslabel</code> ,	
changed to just use		<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\Glsentrysymbolplural</code>	150	<code>\glsinsert</code> and	
changed to just use		<code>\glscustomtext</code>	353
<code>\glsentrysymbolplural</code>	150	<code>\@GLS@:</code> add <code>\glslabel</code> ,	
changed to just use		<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\Glsentrysymbol</code>	149	<code>\glscustomtext</code> and	
changed to just use		<code>\glsinsert</code>	138
<code>\glsentrysymbol</code>	149	change to using <code>\glsentryfmt</code>	
Changed to just use		style commands	138
<code>\Glsentrytext</code>	142		

removed <code>\MakeUppercase</code> (now moved to <code>\glentryfmt</code>) ..	138	<code>\@glsp1@</code> : add <code>\glslabel</code> ,	
<code>\@GLSpl</code> : add <code>\glslabel</code> ,		<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\glsifplural</code> , <code>\glscapscase</code> ,		<code>\glscustomtext</code> and	
<code>\glsinsert</code>	140	<code>\glsinsert</code>	138
change to using <code>\glentryfmt</code>		change to using <code>\glentryfmt</code>	
style commands	140	style commands	138
removed <code>\MakeUppercase</code> as		acronymlists: replaced	
now dealt with in		<code>\@addtoacronymlists</code> with	
<code>\glentryfmt</code>	140	<code>\DeclareAcronymList</code>	22
<code>\@Gls@</code> : add <code>\glsifplural</code> ,		compatible-3.07: new	38
<code>\glscapscase</code> ,		<code>\defglsdisplay</code> : obsoleted	119
<code>\glscustomtext</code> and		<code>\defglsdisplayfirst</code> : obsoleted	119
<code>\glsinsert</code>	137	<code>\defglentryfmt</code> : new	71
change to using <code>\glentryfmt</code>		<code>\forglentries</code> : replaced <code>\ifx</code>	
style commands	137	with <code>\ifdefempty</code>	64
removed <code>\makefirstuc</code> (now		<code>\gls@assign@desc</code> : new	94
dealt with in <code>\glentryfmt</code>)	137	<code>\gls@defglossaryentry</code> : Fixed	
<code>\@Glspl@</code> : add <code>\glsifplural</code> ,		default counter if none	
<code>\glscapscase</code> ,		supplied	98
<code>\glscustomtext</code> and		<code>\gls@doentryfmt</code> : new	71
<code>\glsinsert</code>	139	<code>\glsdisplay</code> : obsoleted	119
change to using <code>\glentryfmt</code>		<code>\glsdisplayfirst</code> : obsoleted ..	119
style commands	139	<code>\glsentryfmt</code> : new	114
removed <code>\makefirstuc</code> (now		<code>\glsgetgrouptitle</code> : Added check	
dealt with in <code>\glentryfmt</code>)	139	in case non-Latin alphabet in	
<code>\@acrlong</code> : added <code>\glslabel</code> ,		use	234
<code>\glsifplural</code> , <code>\glscapscase</code> ,		<code>\gls glossarymark</code> : replaced	
<code>\glsinsert</code> and		<code>\MakeUppercase</code> with	
<code>\glscustomtext</code>	354	<code>\mfirstucMakeUppercase</code> ..	51
<code>\@acrshort</code> : added <code>\glslabel</code> ,		<code>\glsnavigation</code> : switched to	
<code>\glsifplural</code> , <code>\glscapscase</code> ,		using <code>\@gls@getgrouptitle</code>	273
<code>\glsinsert</code> and		<code>\ifglshasdesc</code> : replaced	
<code>\glscustomtext</code>	353	<code>\ifdefempty</code> with	
<code>\@gls@</code> : add <code>\glslabel</code> ,		<code>\ifcempty</code>	67
<code>\glsifplural</code> , <code>\glscapscase</code> ,		<code>\ifglshaslong</code> : new	68
<code>\glscustomtext</code> and		<code>\ifglshasshort</code> : new	68
<code>\glsinsert</code>	136	<code>\ifglshassymbol</code> : replaced	
change to using <code>\glentryfmt</code>		<code>\ifdefempty</code> with	
style commands	136	<code>\ifcempty</code>	68
<code>\@gls@noexpand@fields</code> : Fixed		<code>\ifglsused</code> : replaced	
bug <code>expand</code> replaced with		<code>\ifthenelse</code> with <code>\ifbool</code> ..	65
<code>noexpand</code>	81	<code>\longnewglossaryentry</code> : new ..	94
<code>\@glsdisp</code> : add <code>\glslabel</code> ,		<code>\ns@newglossary</code> : replaced	
<code>\glsifplural</code> , <code>\glscapscase</code> ,		<code>\glsdisplay</code> and	
<code>\glscustomtext</code> and		<code>\glsdisplayfirst</code> with	
<code>\glsinsert</code>	141	<code>\glentryfmt</code>	73
change to using <code>\glentryfmt</code>		numbers: new	40
style commands	141	<code>\SetCustomDisplayStyle</code> :	
		updated to use	
		<code>\defglentryfmt</code>	263

<code>\SetDefaultAcronymDisplayStyle:</code> changed to use <code>\defglentryfmt</code> 260	<code>\gls@assign@desc@field:</code> changed to use <code>\glssetnoexpandfield</code> 24
<code>\SetDescriptionAcronymDisplayStyle:</code> updated to use <code>\defglentryfmt</code> 261	<code>\gls@assign@descplural@field:</code> changed to use <code>\glssetnoexpandfield</code> 25
<code>\SetDescriptionDUAAcronymDisplayStyle:</code> updated to use <code>\defglentryfmt</code> 261	<code>\gls@assign@name@field:</code> changed to use <code>\glssetnoexpandfield</code> 25
<code>\SetDescriptionFootnoteAcronymDisplayStyle:</code> updated to use <code>\defglentryfmt</code> 260	<code>\gls@assign@type@field:</code> changed to use <code>\glssetexpandfield</code> 24
<code>\SetDUADisplayStyle:</code> updated to use <code>\defglentryfmt</code> .. 262	<code>\gls@checkseeallowed:</code> new .. 77
<code>\SetFootnoteAcronymDisplayStyle:</code> updated to use <code>\defglentryfmt</code> 261	<code>\glsaddallunused:</code> set default to <code>\@glo@types</code> 177
<code>\SetSmallAcronymDisplayStyle:</code> updated to use <code>\defglentryfmt</code> 262	<code>\Glsentryfull:</code> changed to use <code>\acrfullformat</code> 174
<code>\setupglossaries:</code> new 42	<code>\glsentryfull:</code> changed to use <code>\acrfullformat</code> 174
<code>\showglolong:</code> new 267	<code>\Glsentryfullpl:</code> changed to use <code>\acrfullformat</code> 174
<code>\showgloshort:</code> new 267	<code>\glsentryfullpl:</code> changed to use <code>\acrfullformat</code> 174
<code>symbols:</code> new 40	<code>\gls glossarymark:</code> renamed <code>\glossarymark</code> to <code>\gls glossarymark</code> to avoid conflict with memoir 51
3.12a – 2013-10-16 <code>\gls@defglossaryentry:</code> added <code>\glslabel</code> 95	<code>\glsprestandardsort:</code> new ... 15
<code>\glsaddkey:</code> new 88	<code>\glssetexpandfield:</code> new 24
3.13a – 2013-11-05 <code>\@gls@assign@symbol@field:</code> changed to use <code>\glssetnoexpandfield</code> 25	<code>\glssetnoexpandfield:</code> new .. 24
<code>\@gls@assign@symbolplural@field:</code> changed to use <code>\glssetnoexpandfield</code> 25	long: switched to <code>\tabularnewline</code> 281
<code>\@gls@link:</code> removed <code>\relax</code> .. 124	long3col: switched to <code>\tabularnewline</code> 282
<code>\@gls@notranslatorhook:</code> new . 29	long3colheader: switched to <code>\tabularnewline</code> 283
<code>\@gls@setupsort@standard:</code> moved <code>\@gls@santizesort</code> to <code>\glsprestandardsort</code> 16	long3colheaderborder: switched to <code>\tabularnewline</code> 283
<code>altsuper4colheader:</code> switched to <code>\tabularnewline</code> 308	long4col: switched to <code>\tabularnewline</code> 284
<code>altsuper4colheaderborder:</code> switched to <code>\tabularnewline</code> 309	long4colheader: switched to <code>\tabularnewline</code> 284
<code>\glossarysection:</code> changed <code>\glossarymark</code> to <code>\gls glossarymark</code> 51	longheader: switched to <code>\tabularnewline</code> 281
	longheaderborder: switched to <code>\tabularnewline</code> 282
	see: added <code>\gls@checkseeallowed</code> 77
	super: switched to <code>\tabularnewline</code> 303

super3col: switched to		\@acrfull: now using	
\tabularnewline	305	\acrfullfmt	243
super3colheader: switched to		\@gls@indexdef: new	41
\tabularnewline	306	\@gls@numbersdef: new	40
super4col: switched to		\@gls@symbolsdef: new	40
\tabularnewline	307	\ACRfullfmt: new	242
super4colheader: switched to		\Acrfullfmt: new	242
\tabularnewline	307	\acrfullfmt: new	242
super4colheaderborder:		\ACRfullplfmt: new	243
switched to		\Acrfullplfmt: new	243
\tabularnewline	308	\acrfullplfmt: new	242
superheader: switched to		\acronymentry: new	246
\tabularnewline	304	dua: new	253
superheaderborder: switched to		dua-desc: new	255
\tabularnewline	304	footnote: new	255
ucmark: added check for memoir	14	footnote-desc: new	257
3.14a – 2013-11-12		footnote-sc: new	256
General: new	327	footnote-sc-desc: new	257
\@glswritefiles: renamed		footnote-sm: new	257
\glswritefiles to		footnote-sm-desc: new	258
\@glswritefiles and used		\Genacrfullformat: new	118
“savewrites” option to set		\genacrfullformat: new	118
\glswritefiles	200	\GenericAcronymFields: new	246
acronyms: new	20	\Genplacrfullformat: new	119
compatible-2.07: added check		\genplacrfullformat: new	118
for 2.07 before setting 3.07		\Glsentryfull: bug fix: added	
compatibility	38	missing \acronymfont	174
\@gls@defglossaryentry: added		\Glsentryfull: bug fix: added	
check for existence of default		missing \acronymfont	174
glossary	96	\Glsentryfullpl: bug fix: added	
set the default for firstplural to		missing \acronymfont	174
be the value of plural	98	\Glsentryfullpl: bug fix: added	
\longprovideglossaryentry:		missing \acronymfont	174
new	95	\glsgenacfmt: new	116
notranslate: new	30	\GlsUseAcrEntryDisplayStyle:	
\provideglossaryentry: new	83	new	248
xindygloss: new	34	\GlsUseAcrStyleDefs: new	248
4.0 – 2013-11-14		index: new	40
\@gls@defglossaryentry: added		long-sc-short: new	250
check for first key	98	long-sc-short-desc: new	251
super: fixed typo in		long-short: new	248
\subglossentry		long-short-desc: new	251
(\glossentrydesc)	303	long-sm-short: new	250
4.01 – 2013-11-16		long-sm-short-desc: new	252
General: fixed non-value options		\newacronymstyle: new	247
so that they can be passed to		numberedsection: added nameref	
document class	11	option	10
4.02 – 2013-12-05		sanitize: fixed bug that caused	
General: Removed		an error here	29
\acronymfont	161–164	sc-short-long: new	250

sc-short-long-desc: new	252	moved \@gls@link@opts and	
\setacronymstyle: new	247	\@gls@link@label to	
\SetGenericNewAcronym: new	245	\@gls@link	123
short-long: new	249	\@gls@writedef: added	
short-long-desc: new	252	\glsdetoklabel	86
sm-short-long: new	251	\@glsdisp: removed \glslabel	
sm-short-long-desc: new	253	(defined in \@gls@link)	141
xindynoglsnumbers: new	34	\@glspl@: removed \glslabel	
4.02 – 2013-13-05		(defined in \@gls@link)	138
\makeglossaries: made		\@printglossary: added	
preamble only	195	\glsdetoklabel	215
4.03 – 2014-01-17		\compatibleglossentry: added	
General: changed default to		\glsdetoklabel	335
\@empty instead of \relax	38	\compatiblesubglossentry:	
4.03 – 2014-01-20		added \glsdetoklabel	335
General: removed \glslabel		dua: fixed bug in \acrfullfmt	254
(defined in \@gls@link)	157	fixed bug in \Acrfullplfmt	254
\@do@esc@wrglossary: added		fixed bug in \acrfullplfmt	254
\glsdetoklabel	207	redefined to use accessibility	
\@do@noesc@wrglossary: added		information	365
\glsdetoklabel	205	dua-desc: commented spurious	
\@ACRlong: removed \glslabel		EOL	255
(defined in \@gls@link)	355	redefined to use accessibility	
\@ACRshort: removed \glslabel		information	367
(defined in \@gls@link)	354	footnote: redefined to use	
\@Acrlong: removed \glslabel		accessibility information	368
(defined in \@gls@link)	355	footnote-desc: redefined to use	
\@Acrshort: removed \glslabel		accessibility information	370
(defined in \@gls@link)	353	footnote-sc: redefined to use	
\@GLS@: removed \glslabel		accessibility information	369
(defined in \@gls@link)	138	footnote-sc-desc: redefined to	
\@GLSpl: removed \glslabel		use accessibility information	370
(defined in \@gls@link)	140	footnote-sm: redefined to use	
\@Gls@: removed \glslabel		accessibility information	370
(defined in \@gls@link)	137	footnote-sm-desc: redefined to	
\@Gls@entry@field: new	165	use accessibility information	370
\@Glspl@: removed \glslabel		\Genacrfullformat: redefined to	
(defined in \@gls@link)	139	use accessibility information	353
\@Acrlong: removed \glslabel		\genacrfullformat: redefined to	
(defined in \@gls@link)	354	use accessibility information	352
\@acrshort: removed \glslabel		\Genplacrfullformat: redefined	
(defined in \@gls@link)	353	to use accessibility	
\@gls@: removed \glslabel		information	353
(defined in \@gls@link)	136	\genplacrfullformat: redefined	
\@gls@access@display: new	344	to use accessibility	
\@gls@entry@field: new	164	information	353
\@gls@fetchfield: new	87	\glossentryname: added	
\@gls@field@link: new	141	\glsdetoklabel	231
\@gls@link: added		\gls@defglossaryentry: added	
\glsdetoklabel	123	\glsdetoklabel	95

long-sm-short-desc: redefined to use accessibility information	364	\showgloplural: added	
\renewacronymstyle: new	248	\glsdetoklabel	264
sc-short-long-desc: redefined to use accessibility information	364	\showglopluralaccess: added	
short-long: commented spurious EOL	250	\glsdetoklabel	372
short-long-desc: redefined to use accessibility information	362	\showgloshort: added	
short-long-desc: redefined to use accessibility information	364	\glsdetoklabel	267
\showglocounter: added		\showgloshortaccess: added	
\glsdetoklabel	265	\glsdetoklabel	373
\showglodesc: added		\showgloshortpluralaccess: added	
\glsdetoklabel	266	\glsdetoklabel	373
\showglodescaccess: added		\showglosort: added	
\glsdetoklabel	372	\glsdetoklabel	266
\showglodescplural: added		\showglosymbol: added	
\glsdetoklabel	266	\glsdetoklabel	266
\showglodescpluralaccess: added		\showglosymbolaccess: added	
\glsdetoklabel	372	\glsdetoklabel	372
\showglofirst: added		\showglosymbolplural: added	
\glsdetoklabel	264	\glsdetoklabel	267
\showglofirstaccess: added		\showglosymbolpluralaccess: added	
\glsdetoklabel	372	\glsdetoklabel	372
\showglofirstpl: added		\showglotext: added	
\glsdetoklabel	264	\glsdetoklabel	264
\showglofirstpluralaccess: added		\showglotextaccess: added	
\glsdetoklabel	372	\glsdetoklabel	372
\showgloflag: added		\showglotype: added	
\glsdetoklabel	267	\glsdetoklabel	264
\showgloindex: added		\showglouserii: added	
\glsdetoklabel	267	\glsdetoklabel	265
\showglolevel: added		\showglouseriii: added	
\glsdetoklabel	264	\glsdetoklabel	265
\showglolong: added		\showglouseriv: added	
\glsdetoklabel	267	\glsdetoklabel	265
\showglolongaccess: added		\showglouservi: added	
\glsdetoklabel	373	\glsdetoklabel	265
\showglolongpluralaccess: added		\showglouserivi: added	
\glsdetoklabel	373	\glsdetoklabel	266
\showglongname: added		sm-short-long-desc: redefined to use accessibility information	365
\glsdetoklabel	266	4.04 – 2014-03-04	
\showglongnameaccess: added		\@gls@getcounterprefix: added	
\glsdetoklabel	372	warning if no prefix can be formed	209
\showgloparent: added		4.04 – 2014-03-06	
\glsdetoklabel	263	General: added sort key to printgloss group	228
		\@@gls@noidx@nosanitizesort: new	26

<code>\@gls@noidx@sanitizesort:</code>		<code>\gls@defglossaryentry:</code>	
new	25	replaced <code>\ifx</code> with	
<code>\@gls@nosanitizesort: new</code>	25	<code>\ifdefvoid</code>	100
<code>\@gls@sanitizesort: new</code>	25	<code>\glsnoidxdisplayloc: new</code>	225
<code>\@glo@addchildren: new</code>	217	<code>\glsnoidxdisplayloclisthandler:</code>	
<code>\@glo@do@sortentries: new</code>	217	new	225
<code>\@glo@grabfirst: new</code>	223	<code>\glsnoidxloclist: new</code>	224
<code>\@glo@sortedinsert: new</code>	218	<code>\glsnoidxloclisthandler: new</code>	225
<code>\@glo@sortentries: new</code>	216	<code>\glsnoidxstripaccents: new</code>	26
<code>\@glo@sorthandler@case: new</code>	219	<code>\makeglossaries: Moved</code>	
<code>\@glo@sorthandler@letter:</code>		definition of <code>\glswrite</code> to	
new	219	<code>\makeglossaries</code>	193
<code>\@glo@sorthandler@nocase:</code>		<code>\makenoidxglossaries: new</code>	195
new	219	<code>\printglossary: changed to use</code>	
<code>\@glo@sorthandler@word: new</code>	218	new <code>\@printglossary</code>	212
<code>\@glo@sortmacro@case: new</code>	220	<code>\printnoidxglossaries: new</code>	213
<code>\@glo@sortmacro@def: new</code>	221	<code>\printnoidxglossary: new</code>	213
<code>\@glo@sortmacro@def@do: new</code>	221	<code>\showgloclist: new</code>	267
<code>\@glo@sortmacro@letter: new</code>	220	<code>\warn@noprintglossary:</code>	
<code>\@glo@sortmacro@nocase: new</code>	220	Activate warning in	
<code>\@glo@sortmacro@standard:</code>		<code>\makeglossaries</code>	212
new	220	<code>\writeist: checked for definition</code>	
<code>\@glo@sortmacro@use: new</code>	221	of <code>\glswrite</code>	180, 184
<code>\@glo@sortmacro@word: new</code>	219	4.06 – 2014-03-12	
<code>\@gls@noidx@do: new</code>	223	General: added	
<code>\@gls@noidx@getgrouptitle:</code>		<code>\glsifhyper</code>	157–164
new	234	<code>\@GLS@: added \glsifhyper</code>	138
<code>\@gls@noref@warn: new</code>	199	<code>\@GLSpl: added \glsifhyper</code>	140
<code>\@gls@reference: new</code>	225	<code>\@GLs@: added \glsifhyper</code>	137
<code>\@gls@warnonglossdefined: new</code>	24	<code>\@GLspl@: added \glsifhyper</code>	139
<code>\@gls@warnonthe glossdefined:</code>		<code>\@gls@: added \glsifhyper</code>	136
new	24	<code>\@gls@numbersdef: added hook</code>	
<code>\@no@makeglossaries: new</code>	199	to set toc title	40
<code>\@print@glossary: new</code>	215	<code>\@gls@symbolsdef: added hook</code>	
<code>\@print@noidx@glossary: new</code>	221	to set toc title	40
<code>\@printgloss@setsort: new</code>	213	<code>\@glsdisp: added \glsifhyper</code>	141
<code>\@printglossary: new</code>	213	<code>\@glspl@: added \glsifhyper</code>	138
<code>alltree: moved hangindent and</code>		acronym: added hook to set toc	
<code>parindent</code> assignments outside		title	20
level test	323	acronyms: added hook to set toc	
<code>\compatibleglossentry: changed</code>		title	20
<code>\newcommand</code> to <code>\def</code> as is		<code>\glsdefmain: added hook to set</code>	
may or may not be defined	335	toc title	19
<code>\compatiblesubglossentry:</code>		4.07 – 2014-04-04	
changed <code>\newcommand</code> to <code>\def</code>		<code>\@glossarysection: added</code>	
as is may or may not be		optional argument when using	
defined	335	unstarred version	52
<code>\glo@grabfirst: new</code>	222	<code>\@gls@noidx@do: added \global</code>	
		in case it's used in a	
		tabular-like style	223

<code>\Acrfullplfmt</code> : fixed no case	
change bug	243
<code>\glsletentryfield</code> : new	165
4.08 – 2014-07-30	
General: added <code>entrycounter</code>	
option to <code>printgloss</code> family	227
added <code>nopostdot</code> option to	
<code>printgloss</code> family	227
added <code>subentrycounter</code> option	
to <code>printgloss</code> family	227
explicitly initialise hyper key	120
moved <code>\glsifhyper</code>	157–164
removed <code>\@sACRlongpl</code>	164
removed <code>\@sAcrlongpl</code>	163
removed <code>\@sacrlongpl</code>	162
removed <code>\@sACRlong</code>	162
removed <code>\@sAcrlong</code>	161
removed <code>\@sacrlong</code>	160
removed <code>\@sACRshortpl</code>	160
removed <code>\@sAcrshortpl</code>	159
removed <code>\@sacrshortpl</code>	158
removed <code>\@sACRshort</code>	158
removed <code>\@sAcrshort</code>	157
removed <code>\@sacrshort</code>	157
removed <code>\@gls@link</code>	122
removed <code>\@sGLSdescplural</code>	148
removed <code>\@sGLsdescplural</code>	148
removed <code>\@sGLsdescplural</code>	148
removed <code>\@sGLSdesc</code>	147
removed <code>\@sGLsdesc</code>	147
removed <code>\@sGLsdesc</code>	147
removed <code>\@sGLsdisp</code>	140
removed <code>\@sGLSfirstplural</code>	145
removed <code>\@sGLsfirstplural</code>	145
removed <code>\@sGLsfirstplural</code>	145
removed <code>\@sGLSfirst</code>	143
removed <code>\@sGLsfirst</code>	143
removed <code>\@sGLsfirst</code>	143
removed <code>\@sGLSname</code>	146
removed <code>\@sGLsname</code>	146
removed <code>\@sGLsname</code>	146
removed <code>\@sGLSplural</code>	144
removed <code>\@sGLsplural</code>	144
removed <code>\@sGLsplural</code>	144
removed <code>\@sGLSpl</code>	140
removed <code>\@sGLspl</code>	139
removed <code>\@sGLspl</code>	138
removed <code>\@sGLSsymbolplural</code>	150
removed <code>\@sGLsymbolplural</code>	150
removed <code>\@sGLsymbolplural</code>	150
removed <code>\@sGLSsymbol</code>	149
removed <code>\@sGLsymbol</code>	149
removed <code>\@sGLsymbol</code>	149
removed <code>\@sGLStext</code>	142
removed <code>\@sGLstext</code>	142
removed <code>\@sGLstext</code>	142
removed <code>\@sGLSuseriii</code>	153
removed <code>\@sGLsuseriii</code>	153
removed <code>\@sGLsuseriii</code>	153
removed <code>\@sGLSuserii</code>	152
removed <code>\@sGLsuserii</code>	152
removed <code>\@sGLsuserii</code>	152
removed <code>\@sGLSuseriv</code>	154
removed <code>\@sGLsuseriv</code>	154
removed <code>\@sGLsuseriv</code>	154
removed <code>\@sGLSuseri</code>	151
removed <code>\@sGLsuseri</code>	151
removed <code>\@sGLsuseri</code>	151
removed <code>\@sGLSuservi</code>	156
removed <code>\@sGLsuservi</code>	156
removed <code>\@sGLsuservi</code>	156
removed <code>\@sGLSuserv</code>	155
removed <code>\@sGLsuserv</code>	155
removed <code>\@sGLsuserv</code>	155
removed <code>\@sGLS</code>	137
removed <code>\@sGLs</code>	137
removed <code>\@sGLs</code>	136
removed <code>\@thirdofthree</code>	
(defined in kernel)	135
removed <code>sPGLS</code>	333
removed <code>sPGLs</code>	331
removed <code>spgls</code>	330
removed <code>sPGLSpl</code>	333
removed <code>sPGLspl</code>	332
removed <code>spglspl</code>	331
<code>\@ACRlong</code> : added	
<code>\do@gls@link@checkfirsthyper</code>	355
<code>\@ACRshort</code> : added	
<code>\do@gls@link@checkfirsthyper</code>	354
<code>\@Acrlong</code> : added	
<code>\do@gls@link@checkfirsthyper</code>	354
<code>\@Acrshort</code> : added	
<code>\do@gls@link@checkfirsthyper</code>	353
<code>\@GLS@</code> : moved <code>\glsifhyper</code>	138
moved check for first use to	
<code>\@gls@link</code>	138

\@GLSp1: moved \glsifhyper ..	140	\ACRfullpl: removed	
moved check for first use to		\s@ACRfullpl	244
\@gls@link	140	switched to using	
\@Gls0: moved \glsifhyper ...	137	\@gls@hyp@opt	244
moved check for first use to		\Acrfullpl: removed	
\@gls@link	137	\s@Acrfullpl	244
\@Glspl0: moved \glsifhyper .	139	switched to using	
moved check for first use to		\@gls@hyp@opt	244
\@gls@link	139	\acrfullpl: removed	
\@acrlong: added		\s@acrfullpl	244
\do@gls@link@checkfirsthyper		switched to using	
.....	354	\@gls@hyp@opt	244
\@acrshort: added		\ACRlong: switched to using	
\do@gls@link@checkfirsthyper		\@gls@hyp@opt	162
.....	353	\Acrlong: switched to using	
\@closegls: new	190	\@gls@hyp@opt	161
\@gls0: moved \glsifhyper ...	136	\acrlong: switched to using	
moved check for first use to		\@gls@hyp@opt	160
\@gls@link	136	\ACRlongpl: switched to using	
\@gls@automake: new	190	\@gls@hyp@opt	164
\@gls@doautomake: new	37	\Acrlongpl: switched to using	
\@gls@field@link: added		\@gls@hyp@opt	163
assignment of		\acrlongpl: switched to using	
\do@gls@link@checkfirsthyper		\@gls@hyp@opt	162
.....	141	\ACRshort: switched to using	
\@gls@forbidtextext: new	71	\@gls@hyp@opt	158
\@gls@hyp@opt: new	121	\Acrshort: switched to using	
\@gls@link: removed		\@gls@hyp@opt	157
redundancy	123	\acrshort: switched to using	
renamed \gls@type to		\@gls@hyp@opt	156
\glstype	123	\ACRshortpl: switched to using	
\@gls@link@checkfirsthyper:		\@gls@hyp@opt	160
new	122	\Acrshortpl: switched to using	
\@glsdisp: moved \glsifhyper	141	\@gls@hyp@opt	159
moved check for first use to		\acrshortpl: switched to using	
\@gls@link	141	\@gls@hyp@opt	158
\@glspl0: moved \glsifhyper .	138	altlongragged4col: fixed bug	
moved check for first use to		that displayed description	
\@gls@link	138	instead of symbol	295
\@ignored@glossaries: new ..	75	automake: new	36
\ACRfull: removed \s@ACRfull	243	\forallacronyms: new	63
switched to using		\GLS: switched to using	
\@gls@hyp@opt	243	\@gls@hyp@opt	137
\Acrfull: removed \s@Acrfull	243	\Gls: switched to using	
switched to using		\@gls@hyp@opt	136
\@gls@hyp@opt	243	\gls: switched to using	
\acrfull: removed \s@acrfull	243	\@gls@hyp@opt	136
switched to using		\gls@defglossaryentry: added	
\@gls@hyp@opt	243	check for ignored glossary ..	96
		\gls@istfilebase: new	48

\glsaddkey: removed		\glsname: switched to using	
\@sGLS@user@(<i>key</i>)	90	\@gls@hyp@opt	145
removed \@sGls@user@(<i>key</i>)	90	\GLSpl: switched to using	
removed \@sgls@user@(<i>key</i>)	89	\@gls@hyp@opt	140
switched to using		\Glspl: switched to using	
\@gls@hyp@opt	89, 90	\@gls@hyp@opt	139
\GLSdesc: switched to using		\glspl: switched to using	
\@gls@hyp@opt	147	\@gls@hyp@opt	138
\Glsdesc: switched to using		\GLSplural: switched to using	
\@gls@hyp@opt	147	\@gls@hyp@opt	144
\glsdesc: switched to using		\Glsplural: switched to using	
\@gls@hyp@opt	146	\@gls@hyp@opt	144
\GLSdescplural: switched to		\glsplural: switched to using	
using \@gls@hyp@opt	148	\@gls@hyp@opt	143
\Glsdescplural: switched to		\glspace: new	242
using \@gls@hyp@opt	148	\GLSsymbol: switched to using	
\glsdescplural: switched to		\@gls@hyp@opt	149
using \@gls@hyp@opt	147	\Glsymbol: switched to using	
\glsdisablehyper: added		\@gls@hyp@opt	149
\KV@glslink@hyperfalse to		\glsymbol: switched to using	
definition	135	\@gls@hyp@opt	148
\glsdisp: switched to using		\GLSsymbolplural: switched to	
\@gls@hyp@opt	140	using \@gls@hyp@opt	150
\glsdohyperlink: new	134	\Glsymbolplural: switched to	
\glsdohypertarget: new	134	using \@gls@hyp@opt	150
\glsenablehyper: added		\glsymbolplural: switched to	
\KV@glslink@hypertrue to		using \@gls@hyp@opt	149
definition	135	\GLStext: switched to using	
\GLSfirst: switched to using		\@gls@hyp@opt	142
\@gls@hyp@opt	143	\Glstext: switched to using	
\Glsfirst: switched to using		\@gls@hyp@opt	142
\@gls@hyp@opt	143	\glstext: switched to using	
\glsfirst: switched to using		\@gls@hyp@opt	142
\@gls@hyp@opt	143	\glstreenamefmt: new	316
\GLSfirstplural: switched to		\GLSuseri: switched to using	
using \@gls@hyp@opt	145	\@gls@hyp@opt	151
\Glsfirstplural: switched to		\Glsuseri: switched to using	
using \@gls@hyp@opt	145	\@gls@hyp@opt	151
\glsfirstplural: switched to		\glsuseri: switched to using	
using \@gls@hyp@opt	144	\@gls@hyp@opt	150
\glsifhyper: deprecated	121	\GLSuserii: switched to using	
\glslink: switched to using		\@gls@hyp@opt	152
\@gls@hyp@opt	122	\Glsuserii: switched to using	
\glslinkcheckfirsthyperhook:		\@gls@hyp@opt	152
new	123	\glsuserii: switched to using	
\glslinkvar: new	121	\@gls@hyp@opt	151
\GLSname: switched to using		\GLSuseriii: switched to using	
\@gls@hyp@opt	146	\@gls@hyp@opt	153
\Glsname: switched to using		\Glsuseriii: switched to using	
\@gls@hyp@opt	146	\@gls@hyp@opt	153

<code>\glsuseriii</code> : switched to using <code>\@gls@hyp@opt</code>	152	<code>\@gls@glossary</code> : Renamed <code>\@glossary</code> to <code>\@gls@glossary</code>	201
<code>\GLSuseriv</code> : switched to using <code>\@gls@hyp@opt</code>	154	<code>alttree</code> : moved space after symbol	323, 324
<code>\Glsuseriv</code> : switched to using <code>\@gls@hyp@opt</code>	154	<code>\glspercentchar</code> : new	178
<code>\glsuseriv</code> : switched to using <code>\@gls@hyp@opt</code>	153	<code>\glstildechar</code> : new	178
<code>\GLSuserv</code> : switched to using <code>\@gls@hyp@opt</code>	155	4.11 – 2014-09-01 <code>\@do@esc@wrglossary</code> : added hook	207
<code>\Glsuserv</code> : switched to using <code>\@gls@hyp@opt</code>	155	<code>\gls@wrglossary</code> : renamed from <code>\@wrglossary</code> to <code>\gls@wrglossary</code>	202
<code>\glsuserv</code> : switched to using <code>\@gls@hyp@opt</code>	154	<code>\glsaddprotectedpagefmt</code> : new	203
<code>\GLSuservi</code> : switched to using <code>\@gls@hyp@opt</code>	156	<code>\glsbackslash</code> : new	178
<code>\Glsuservi</code> : switched to using <code>\@gls@hyp@opt</code>	156	<code>sanitize</code> : none option	29
<code>\glsuservi</code> : switched to using <code>\@gls@hyp@opt</code>	155	4.12 – 2014-11-22 <code>\@gls@addpredefinedattributes</code> : Added <code>glsignore</code> attribute . .	57
<code>\ifignoredglossary</code> : new	75	<code>\@gls@adjustmode</code> : new	177
<code>\newglossary</code> : added starred version	72	<code>\@gls@notranslatorhook</code> : removed	29
<code>\newignoredglossary</code> : new . . .	74	<code>\@gls@toc</code> : added <code>\protect</code> to <code>\numberline</code>	54
<code>\ns@newglossary</code> : added <code>\@glotype@{<i>name</i>}@log</code>	72	<code>\@gls@usetranslator</code> : new	29
new	72	<code>\glsacrpluralsuffix</code> : new	45
<code>\p@gls@hyp@opt</code> : new	121	<code>\glsadd</code> : added check for vertical mode	176
<code>\PGLS</code> : changed to use <code>\@gls@hyp@opt</code>	333	<code>\glsaddallunused</code> : replaced <code>@gobble</code> with <code>glsignore</code> . . .	177
<code>\Pgls</code> : changed to use <code>\@gls@hyp@opt</code>	331	<code>\glsifusedtranslatordict</code> : new	29
<code>\ppls</code> : changed to use <code>\@gls@hyp@opt</code>	330	<code>\glsignore</code> : new	177
<code>\PGLSpl</code> : changed to use <code>\@gls@hyp@opt</code>	333	<code>\glsupacrpluralsuffix</code> : new . .	45
<code>\Pglspl</code> : changed to use <code>\@gls@hyp@opt</code>	332	<code>\ProvidesGlossariesLang</code> : new	45
<code>\pplspl</code> : changed to use <code>\@gls@hyp@opt</code>	331	<code>\RequireGlossariesLang</code> : new .	45
<code>\s@gls@hyp@opt</code> : new	121	4.13 – 2015-02-03 <code>\indexspace</code> : new	276, 297, 316
<code>\s@newglossary</code> : new	72	4.14 – 2015-02-28 <code>\@@glslocalreset</code> : new	106
4.09 – 2014-08-12 <code>\glsaddkey</code> : fixed bug in user commands	89	<code>\@@glslocalunset</code> : new	106
4.10 – 2014-08-27 <code>\@Gls@acrentryname</code> : new	166	<code>\@@glsreset</code> : new	107
<code>\@Gls@entryname</code> : new	166	<code>\@@glsunset</code> : new	106
		<code>\@@newglossaryentry@defcounters</code> : new	108
		<code>\@cGls</code> : new	111
		<code>\@cGls@</code> : new	111
		<code>\@cGlspl@</code> : new	112
		<code>\@cgls</code> : new	111
		<code>\@cgls@</code> : new	111
		<code>\@cglspl</code> : new	112

\cglspl@: new	112	\@GLSpl: added	
\@gls@entry@count: new	111	\glspostlinkhook	140
\@gls@increment@currcount:		\@Gls@: added	
new	110	\glspostlinkhook	137
\@gls@local@increment@currcount:		\@GLspl@: added	
new	110	\glspostlinkhook	139
\@gls@write@entrycounts: new	110	\@acrlong: added	
\@glslocalreset: new	106	\glspostlinkhook	354
\@glslocalunset: new	106	\@acrshort: added	
\@glsreset: new	106	\glspostlinkhook	353
\@glsunset: new	106	\@gls@: added	
\@newglossaryentry@defcounters:		\glspostlinkhook	136
new	101	\@gls@@link: added	
\cGls: new	111	\glspostlinkhook	122
\cgl: new	111	\@gls@field@link: added	
\cGlsformat: new	112	\glspostlinkhook	141
\cglformat: new	111	\@gls@link: moved definition of	
\cGLspl: new	112	\glsifhyperon outside of this	
\cglspl: new	112	macro	124
\cGlsplformat: new	113	\@glsdisp: added	
\cglsplformat: new	112	\glspostlinkhook	141
\gls@defdocnewglossaryentry:		\@GLspl@: added	
new	83	\glspostlinkhook	139
\glsenableentrycount: new	108	\glsacspace: new	249
\glslocalreset: switched to		\glsadd: changed	
\@glslocalreset	105	\@do@wrglossary to	
\glslocalunset: switched to		\@do@wrglossary	176
\@glslocalunset	106	\glsfielddef: new	92
\glsreset: switched to		\glsfieldedef: new	91
\@glsreset	105	\glsfieldfetch: new	92
\glsunset: switched to		\glsfieldgdef: new	91
\@glsunset	105	\glsfieldxdef: new	91
4.15 – 2015-03-16		\glsifhyperon: moved definition	
General: bug fix replaced		of \glsifhyperon	123
\@glo@type with \glstype	163	\glslinkpostsetkeys: new	123
4.16 – 2015-06-18		\glspostlinkhook: new	122
\glsaddstoragekey: new	87	\glswriteentry: new	202
4.16 – 2015-07-08		\ifglsfieldcseq: new	94
General: added		\ifglsfielddefeq: new	93
\glspostlinkhook	157–164	\ifglsfieldeq: new	93
\@ACRlong: added		long-sp-short: new	249
\glspostlinkhook	355	long-sp-short-desc: new	251
\@ACRshort: added		\showglofield: new	268
\glspostlinkhook	354	4.18 – 2015-09-09	
\@Acrlong: added		General: split mfirstuc into	
\glspostlinkhook	355	separate bundle	4
\@Acrshort: added		4.19 – 2015-10-31	
\glspostlinkhook	354	\glstreenamebox: new	322
\@GLS@: added			
\glspostlinkhook	138		

4.19 – 2015-11-22			
\@gls@link@nocheckfirsthyper:		\glsentrytitlecase: new	170
new	141	\glsfindwidesttoplevelname:	
\@gls@preglossaryhook: new . .	213	new	322
\@printglossary: added		\glslistgroupheaderfmt: new .	276
\@gls@preglossaryhook . .	215	\glslistnavigationitem: new .	276
\do@glsdisablehyperinlist:		\glstreegroupheaderfmt: new .	316
new	123	\glstreenavigationfmt: new .	316
\doifglossarynoexistsordo:		\ifglswrallowprimitivemods:	
new	66	new	205
\gls@gobbleopt: new	72	list: fixed missing space before	
\glsdoifexistsordo: new	66	description	277
4.20 – 2015-11-30		long: fixed typo in	
\@gls@link: added		\glossentrydesc	281
\@gls@setdefault@glslink@opts		mcolalttreesspannav: new	302
.	124	mcolindexspannav: new	298
added \glsdonohyperlink		mcoltreenonamespannav: new .	300
when hyperlink is suppressed	124	mcoltreesspannav: new	299
\@gls@setdefault@glslink@opts:		super4col: fixed bug in	
new	123	\glossentry	307
\gls@checkseeallowed@preambleonly:		4.23 – 2016-04-30	
new	78	altlongragged4col: check for	
\glsdonohyperlink: new	134	nogroupskip changed	295
4.21 – 2016-01-24		altsuperragged4col: check for	
General: changed checkfirsthyper		nogroupskip changed	314
assignment	157–164	\glscurrentfieldvalue: new .	70
\@printglossary: warn if no		\ifglshasfield: added	
style has been set	213	\glscurrentfieldvalue	69, 70
altlong4col-booktabs: new . .	288	long: check for nogroupskip	
altlongragged4col-booktabs:		changed	281
new	289	long-booktabs: check for	
\glsLTpenaltycheck: new	290	nogroupskip changed	287
\glspatchLoutput: new	290	long3col: check for nogroupskip	
\glspenaltygroupskip: new . .	290	changed	282
long-booktabs: new	287	long3col-booktabs: check for	
long3col-booktabs: new	287	nogroupskip changed	287
long4col-booktabs: new	288	long4col: check for nogroupskip	
longragged-booktabs: new . . .	289	changed	284
longragged3col-booktabs: new	289	long4col-booktabs: check for	
\setglossarystyle: set default		nogroupskip changed	288
style if not already set	235	longragged: check for	
4.22 – 2016-04-19		nogroupskip changed	292
\@@do@esc@wrglossary: added		longragged3col: check for	
check for \@arabic	206	nogroupskip changed	294
added test to allow temporary		super: check for nogroupskip	
primitive modifications and		changed	303
added arabic case	206	super3col: check for nogroupskip	
\gls@arabicpage: new	203	changed	305
\gls@protected@pagefmts:		super4col: check for nogroupskip	
added arabic to list	202	changed	307

superragged: check for nogroupskip changed	311	outside of theglossary environment	222
superragged3col: check for nogroupskip changed	313	\glsnavhyperlinkname: new	270
4.24 – 2016-05-27		4.30 – 2017-06-11	
\@gls@extramakeindexopts: new	188	\@glo@autosee: new	100
\@gls@glossary: added check for debug mode	201	\@glo@autoseehook: new	101
\@gls@see@noindex: new	9	\@glo@check@sortallowed: new	16
debug: new	7	\@gls@noidx@do: letter group assignment made global	224
\glsnomakeindexwarning: new	54	\@gls@setupsort@def: added check for register	17
\GlsSetQuote: new	185	\@gls@setupsort@none: new	18
\GlsSetWriteIstHook: new	185	\@xdycrossrefhook: new	59
seenoindex: new	9	\@xdylocationclassorder: bug fix: changed \edef to \def	60
4.25 – 2016-06-09		\glosortentrieswarning: new	23
General: changed		\gls@set@xr@key: new	77
\DeclareRobustCommand to \newrobustcmd and changed		\gls@xr@key: new	77
\@ifundefined to \ifcsundef	347	\GlsAddXdyLocation: bug fix: changed #1 to #2	60
\@gls@enablesavenonumberlist: new	79	\glsnoidxstripaccents: added \a	26
\@gls@initnonumberlist: new	78, 79	added \TH, \dh and \DH	27
\@gls@savenonumberlist: new	78	4.31 – 2017-08-10	
4.26 – 2016-10-12		nolist: added check for “list” style	13
\@glossary@default@style: added check for classicthesis	11	4.31 – 2017-09-10	
\glstreechildpredesc: new	317	style: changed \renewcommand to \def	11
\glstreeitem: new	317	4.32 – 2017-08-24	
\glstreepredesc: new	317	\@glsnavhypertarget: new	272
\glstreesubitem: new	317	\@glsshowtarget: new	8
\glstreesubsubitem: new	317	\glsshowtarget: new	8
mcolindex: replaced \@idxitem with \glstreeitem	297	4.33 – 2017-09-20	
mcolindexspannav: replaced \@idxitem with \glstreeitem	298	\@do@esc@wrglossary: added \gls@the and \gls@number	207
4.28 – 2017-01-07		renamed from \@do@esc@wrglossary	205
\glspatchtabularx: new	105	\@do@noesc@wrglossary: new	204
4.29 – 2017-01-19		\@do@wrglossary: changed to check for eslocations	204
General: added check for \@glsxtr@doaccsupp	335	\@gls@missinglang@warn: new	23
\@gls@noidx@do: current letter group assignment made global	224	eslocations: new	12
\@print@noidx@glossary: moved definition of \@gls@currentlettergroup		\GlsSetXdyFirstLetterAfterDigits: added starred version	178
		\GlsSetXdyNumberGroupOrder: new	179

4.34 – 2017-11-03		
<code>\glsshowtarget</code> : modified to check for math mode and inner	8	
<code>mcolaltreespannav</code> : removed spurious space	302	
4.35 – 2017-11-14		
<code>\glsadd</code> : added <code>\@gls@setsort</code> (in case of <code>sort=use</code>)	176	
4.36 – 2018-03-07		
<code>\@gls@glossary</code> : removed <code>\index</code>	201	
4.37 – 2018-04-07		
<code>\gls@begindocdefs</code> : new	85	
4.38 – 2018-05-10		
<code>\@gls@define@glossaryentrycounter</code> : added check for existence of glossaryentry counter	14	
new	14	
<code>\@gls@define@glossarysubentrycounter</code> : new	15	
prepended <code>\currentglossary.</code> to <code>\theHglossarysubentry</code> and removed spurious eol space	15	
<code>\glsaccsupp</code> : added braces around actual text argument	344	
<code>\glsentrycounterlabel</code> : bug fix: move conditional inside command	230	
<code>\GlsEntryCounterLabelPrefix</code> : new	227	
<code>\glsentryitem</code> : bug fix: move conditional inside command	230	
<code>\glsrefentry</code> : bug fix: move conditional inside command	229	
<code>\glsresetentrycounter</code> : bug fix: move conditional inside command	229	
<code>\glsresetsubentrycounter</code> : bug fix: move conditional inside command	229	
<code>\glsstepentry</code> : bug fix: move conditional inside command	229	
<code>\glsstepsubentry</code> : bug fix: move conditional inside command	229	
<code>\glssubentrycounterlabel</code> : bug fix: move conditional inside command	230	
		<code>\glssubentryitem</code> : bug fix: move conditional inside command
		<code>\showglnonameaccess</code> : bug fix: corrected field (was showing text access field)
		4.40 – 2018-06-01
		<code>\istfile</code> : changed <code>\def</code> to <code>\providecommand</code>
		<code>\makenoidxglossaries</code> : false
		4.41 – 2018-07-23
		General: changed <code>\val</code> and <code>\nr</code> to <code>\gls@numberedsection@val</code> and <code>\gls@numberedsection@nr</code>
		<code>\@gls@override@glossary</code> : new
		debug: changed <code>\val</code> and <code>\nr</code> to <code>\gls@debug@val</code> and <code>\gls@debug@nr</code>
		<code>\glossary</code> : added warning
		<code>\gls@original@glossary</code> : new
		<code>\gls@original@makeglossary</code> : new
		<code>kernelglossredefs</code> : new
		<code>\makeglossaries</code> : removed redefinition of <code>\makeglossary</code>
		<code>\makeglossary</code> : added warning
		<code>nonumberlist</code> : changed <code>\val</code> and <code>\nr</code> to <code>\gls@nonumberlist@val</code> and <code>\gls@nonumberlist@nr</code>
		<code>numberedsection</code> : changed <code>\val</code> and <code>\nr</code> to <code>\gls@numberedsection@val</code> and <code>\gls@numberedsection@nr</code>
		<code>seenoindex</code> : changed <code>\val</code> and <code>\nr</code> to <code>\gls@seenoindex@val</code> and <code>\gls@seenoindex@nr</code>
		<code>translate</code> : changed <code>\val</code> and <code>\nr</code> to <code>\gls@translate@val</code> and <code>\gls@translate@nr</code>
		4.42 – 2019-01-06
		<code>\@gls@@automake@immediate</code> : new
		<code>\@gls@automake@immediate</code> : new
		<code>altlong4col-booktabs</code> : removed superfluous <code>\glspatchLOutput</code>

automake: switch from boolean to choice	36	\@GLSuserv@: added accessibility support	360
\gls@automake@nr: new	36	\@GLSuserii@: added accessibility support	359
\glsfielddedef: changed from \edef to \protected@csedef	91	\@GLSuseriii@: added accessibility support	359
\glsfieldxdef: changed from \edef to \protected@csxdef	91	\@GLSuseriv@: added accessibility support	359
\ifglsautomake: now defined explicitly instead of through boolean key	36	\@GLSdesc@: added accessibility support	357
noglossaryindex: new	41	\@GLSdescplural@: added accessibility support	357
4.43 – 2019-09-28		\@GLSfirst@: added accessibility support	356
\glsnoidxstripaccents: add check for LaTeX version 2019/10/01	27	\@GLSfirstplural@: added accessibility support	356
4.44 – 2019-12-06		\@GLSname@: added accessibility support	357
\glsprefix@record@hook: new	330	\@GLSplural@: added accessibility support	356
4.45 – 2020-02-13		\@GLSsymbol@: added accessibility support	358
General: removed backward compatibility use of symbol key	338	\@GLSsymbolplural@: added accessibility support	358
\@do@write@glslabels: new	35	\@GLStext@: added accessibility support	355
\@glsshowtarget: new	8	\@GLSuseri@: added accessibility support	358
\@GLSdesc@: added accessibility support	357	\@GLSuserii@: added accessibility support	359
\@GLSdescplural@: added accessibility support	357	\@GLSuseriii@: added accessibility support	359
\@GLSfirst@: added accessibility support	356	\@GLSuseriv@: added accessibility support	359
\@GLSfirstplural@: added accessibility support	356	\@GLSuserv@: added accessibility support	360
\@GLSname@: added accessibility support	357	\@GLSuservi@: added accessibility support	360
\@GLSplural@: added accessibility support	356	\@do@write@glslabels: new	35
\@GLSsymbol@: added accessibility support	358	\@domakeglossaries: new	34
\@GLSsymbolplural@: added accessibility support	358	\@gls@access@display: deprecated	344
\@GLStext@: added accessibility support	355	\@gls@fieldaccess@display: new	344
\@GLSuseri@: added accessibility support	358	\@glsdesc@: added accessibility support	357
\@GLSuserii@: added accessibility support	359	\@glsdescplural@: added accessibility support	357
\@GLSuseriii@: added accessibility support	359	\@glsfirst@: added accessibility support	356
\@GLSuseriv@: added accessibility support	359	\@glsfirstplural@: added accessibility support	356
		\@glsname@: added accessibility support	357

\@glsplural@: added accessibility support	356	\Glsentryprefixfirstplural: added \glsdetoklabel	329
\@glsshowaccsupp: new	9	\glsentryprefixfirstplural: added \glsdetoklabel	328
\@glssymbol@: added accessibility support	358	\Glsentryprefixplural: added \glsdetoklabel	329
\@glssymbolplural@: added accessibility support	358	\glsentryprefixplural: added \glsdetoklabel	328
\@glstext@: added accessibility support	355	\glsentrytitlecase: added existence check	170
\@glsuseri@: added accessibility support	358	\glsentryuseriaccess: new	342
\@glsuserii@: added accessibility support	359	\glsentryuseriiaccess: new	342
\@glsuseriii@: added accessibility support	359	\glsentryuseriiiaccess: new	342
\@glsuseriv@: added accessibility support	359	\glsentryuserivaccess: new	342
\@glsuserv@: added accessibility support	360	\glsentryuserviaccess: new	342
\@glsuservi@: added accessibility support	360	\glsfieldaccsupp: new	343
\@org@Gls@acrentryname: added check for \glsshortaccessdisplay	167	\glsfirstaccessdisplay: added check for existence	345
acronyms: changed \renewcommand to \def	20	\glsfirstpluralaccessdisplay: added check for existence	345
debug: showaccsupp	7	\glslongaccessdisplay: added check for existence	346
disablemakegloss: new	34	\glslongpluralaccessdisplay: added check for existence	346
\gls@accessibility: new	335	\glsnameaccessdisplay: added check for existence	344
\gls@accsupp@engine: new	335	\glspluralaccessdisplay: added check for existence	345
\glsaccessibility: new	335	\glsprefixsep: new	330
\glsdefaultshortaccess: new	371	\glssee: switched to \newrobustcmd	210
\glsdescriptionaccessdisplay: added check for existence	345	\glsseeformat: switched to \newrobustcmd	210
\glsdescriptionpluralaccessdisplay: added check for existence	346	\glsseeitem: switched to \newrobustcmd	211
\glsentrydescpluralaccess: corrected field reference	341	\glsseelist: switched to \newrobustcmd	210
\glsentryfirstpluralaccess: switched to using \@gls@entry@field	341	\glsshortaccessdisplay: added check for existence	346
\glsentryparent: new	171	\glsshortaccsupp: new	344
\Glsentryprefix: added \glsdetoklabel	329	\glsshortplaccsupp: new	344
\glsentryprefix: added \glsdetoklabel	328	\glsshortpluralaccessdisplay: added check for existence	346
\Glsentryprefixfirst: added \glsdetoklabel	328	\glsshowaccsupp: new	9
\glsentryprefixfirst: added \glsdetoklabel	328	\glsshowtargetfont: new	8
		\glsshowtargetouter: new	8
		\glsshowtargetsymbol: new	8
		\glsymbolaccessdisplay: added check for existence	345

<code>\glssymbolpluralaccessdisplay:</code>		<code>\ifglossaryexists:</code> added	
added check for existence	345	starred form	64
<code>\glstextaccessdisplay:</code> added		<code>\s@ifglossaryexists:</code> new	64
check for existence	345	<code>\setglossarypreamble:</code> switched	
<code>\glsuseriaccessdisplay:</code> new	346	to starred form of	
<code>\glsuseriiaccessdisplay:</code> new	346	<code>\ifglossaryexists</code>	50
<code>\glsuseriiiaccessdisplay:</code>		4.47 – 2021-09-20	
new	347	General: changed <code>\x</code> to <code>\@gls@x</code>	238
<code>\glsuserivaccessdisplay:</code> new	347	<code>\@do@esc@wrglossary:</code> changed	
<code>\glsuservaccessdisplay:</code> new	347	<code>\edef</code> to <code>\protected@edef</code>	207
<code>\glsuserviiaccessdisplay:</code> new	347	<code>\@do@noesc@wrglossary:</code>	
<code>\ifglshaschildren:</code> made robust	67	changed <code>\edef</code> to	
<code>\ifglshasfield:</code> made robust	69	<code>\protected@edef</code>	205
<code>\ifglshaslong:</code> made robust	68	<code>\@do@write@glslabels:</code> fully	
<code>\ifglshasprefix:</code> added		expand list	35
<code>\glsdetoklabel</code>	329	<code>\@gls@@default@acr@descplural:</code>	
<code>\ifglshasprefixfirst:</code> added		new	260
<code>\glsdetoklabel</code>	329	<code>\@gls@@default@acr@firstplural:</code>	
<code>\ifglshasprefixfirstplural:</code>		new	260
added <code>\glsdetoklabel</code>	330	<code>\@gls@@default@acr@plural:</code>	
<code>\ifglshasprefixplural:</code> added		new	260
<code>\glsdetoklabel</code>	329	<code>\@gls@noexpand@assign@default@plurals:</code>	
<code>\ifglshasshort:</code> made robust	68	new	260
<code>\ifglshassymbol:</code> made robust	68	<code>\glo@do@sortentries:</code> changed	
<code>\makeglossaries:</code> let		<code>\edef</code> to <code>\protected@edef</code>	218
<code>\@makeglossary</code> to <code>\@gobble</code>		<code>\glo@sortentries:</code> changed	
instead of <code>\relax</code>	194	<code>\edef</code> to <code>\protected@edef</code>	217
<code>restoremakeloss:</code> new	34	<code>\glo@storeentry:</code> changed	
<code>user1access:</code> new	337	<code>\edef</code> to <code>\protected@edef</code>	102
<code>user2access:</code> new	337	<code>\@gls@checkesc:</code> new	132
<code>user3access:</code> new	337	<code>\@gls@deprecated@acr@warning:</code>	
<code>user4access:</code> new	338	new	262
<code>user5access:</code> new	338	<code>\@gls@link:</code> changed <code>\edef</code> to	
<code>user6access:</code> new	338	<code>\protected@edef</code>	123
<code>writeglslabels:</code> new	36	<code>\@gls@setupsort@use:</code> changed	
<code>\xglfieldaccsupp:</code> new	343	<code>\edef</code> to <code>\protected@edef</code>	18
4.46 – 2020-03-19		<code>\@gls@writedef:</code> added a	
<code>\@ifglossaryexists:</code> new	64	warning if entry has already	
<code>\@printgloss@checkexists:</code>		been defined	86
new	212	<code>\@makeglossaries@warn@noprintglossary:</code>	
<code>\@printgloss@checkexists@allowignored:</code>		new	195
new	212	<code>\@write@glslabelsname:</code> new	35
<code>\@printgloss@checkexists@noignored:</code>		<code>\do@write@glslabelsname:</code> new	35
new	212	<code>\forglslentries:</code> changed <code>\edef</code>	
<code>\@printglossary:</code> replaced		to <code>\protected@edef</code>	64
<code>\ifglossaryexists</code> with		<code>\gls@defdocnewglossaryentry:</code>	
<code>\@printgloss@checkexists</code>	214	removed see warning	83
<code>\doifglossarynoexistsordo:</code>		<code>\gls@defglossaryentry:</code> changed	
switched to starred form of		<code>\edef</code> to	
<code>\ifglossaryexists</code>	66	<code>\protected@edef</code>	95, 96

<code>\gls@docdef@addlabel</code> : new ..	84	<code>\SetDescriptionFootnoteAcronymStyle</code> :	
<code>\gls@docdef@labellist</code> : new ..	84	added deprecated warnings	261
<code>\gls@glsdefs@addlabel</code> : new ..	84	<code>\SetDUASStyle</code> : added deprecated	
<code>\gls@glsdefs@labellist</code> : new ..	84	warnings	262
<code>\gls@save@numberlist</code> : changed		<code>\SetFootnoteAcronymStyle</code> :	
<code>\edef</code> to <code>\protected@edef</code>	211	added deprecated warnings	261
<code>\glsadd</code> : changed <code>\edef</code> to		<code>\SetSmallAcronymStyle</code> : added	
<code>\protected@edef</code>	176	deprecated warnings	262
<code>\glsaddall</code> : changed <code>\edef</code> to		<code>writeglslabelnames</code> : new	36
<code>\protected@edef</code>	177	4.48 – 2021-10-19	
<code>\glsdefs@newdocentry</code> : new ..	85	<code>\glscapitalisewords</code> : new	170
<code>\glsdisplaynumberlist</code> : changed		<code>\glslistexpandedname</code> : new	277
<code>\edef</code> to <code>\protected@edef</code>	175	<code>\glslistinit</code> : new	276
<code>\glsfielddef</code> : changed <code>\edef</code> to		<code>\glsunexpandedfieldvalue</code> : new	92
<code>\protected@edef</code>	92	4.49 – 2021-11-01	
<code>\glsfielddedef</code> : changed <code>\edef</code> to		<code>\glslistinit</code> : added	
<code>\protected@edef</code>	91	<code>\glsnavigation</code>	276
<code>\glsfieldfetch</code> : changed <code>\edef</code>		4.50 – 2022-10-14	
to <code>\protected@edef</code>	92	General: added initialisation	240
<code>\glsfieldgdef</code> : changed <code>\edef</code> to		allow markup to be stripped	
<code>\protected@edef</code>	91	from location	238
<code>\glsfieldxdef</code> : changed <code>\edef</code> to		corrected description mapping	338
<code>\protected@edef</code>	91	don't load <code>glossary-super</code> if <code>xtab</code>	
<code>\glsIfListOfAcronyms</code> : changed		has been loaded	12
<code>\edef</code> to <code>\protected@edef</code> ..	21	No longer requires textcase	
<code>\glsmoveentry</code> : changed <code>\edef</code> to		with <code>mfirstuc 2.08+</code>	5
<code>\protected@edef</code>	101	removed automatic loading of	
<code>\glsnavhyperlink</code> : changed		<code>glossaries-compatible-307</code>	42
<code>\edef</code> to <code>\protected@edef</code>	270	switched to using	
<code>\GlsSetQuote</code> : added check for		<code>\glsuppercase</code>	
<code>backslash</code>	187 142–156, 158, 160, 162, 164	
<code>\glsstepsubentry</code> : changed		<code>\@do@esc@wrglossary</code> : moved	
<code>\edef</code> to <code>\protected@edef</code>	229	<code>\romannumeral</code> inside allow	
<code>\ifglsfieldcseq</code> : changed <code>\edef</code>		primitive mods	207
to <code>\protected@edef</code>	94	<code>\@gls@@default@acr@descplural</code> :	
<code>\ifglshaschildren</code> : changed		removed	260
<code>\edef</code> to <code>\protected@edef</code> ..	67	<code>\@gls@@default@acr@firstplural</code> :	
<code>\ifignoredglossary</code> : changed		removed	260
<code>\edef</code> to <code>\protected@edef</code> ..	75	<code>\@gls@@default@acr@plural</code> :	
<code>\makenoidxglossaries</code> : changed		removed	260
<code>\edef</code> to <code>\protected@edef</code>	196	<code>\@gls@default@entryfmt</code> :	
<code>\new@glossaryentry</code> : allow		removed	347
overwrite if defined in		<code>\@gls@navhypertarget</code> : new ..	271
<code>.glsdefs</code>	84	<code>\@gls@noexpand@assign@default@plurals</code> :	
<code>\newignoredglossary</code> : changed		removed	260
<code>\edef</code> to <code>\protected@edef</code> ..	74	<code>\@Gls@acronymname</code> : removed	
<code>\ns@newglossary</code> : changed <code>\edef</code>		<code>mfirstuc</code> internal commands	166
to <code>\protected@edef</code>	72	<code>\@Glsdisp</code> : new	141
<code>\SetDescriptionDUAAcronymStyle</code> :		<code>\@Glsentryfield</code> : new	165
added deprecated warning ..	261		

\@domakeglossaries: added		\CustomNewAcronymDef:	
check for \nofiles	34	removed	263
\@gls@automake: added check for		\DefaultNewAcronymDef:	
void codepage	190	removed	260, 371
\@gls@automake@immediate:		\defglsdisplay: removed	119
added check for void		\defglsdisplayfirst: removed	119
codepage	191	\DescriptionDUANewAcronymDef:	
\@gls@automake@types: new	192	removed	261
\@gls@defaulttype: new	71	\DescriptionFootnoteNewAcronymDef:	
\@gls@do@automake@err: new	36	removed	260, 371
\@gls@do@glsunset: new	120	\DescriptionNewAcronymDef:	
\@gls@entry@field: switched to		removed	261, 371
\csuse	164	dua: switched to using	
\@gls@escbsdq: check		\glsuppercase	253, 254
esclocations setting	127	dua-desc: removed \acronymfont	
\@gls@getcounterprefix:		from around long form	255
changed \edef to \def	209	\DUANewAcronymDef: removed	262
\@gls@getgrouptitle: switched		esclocations: default now false	12
to L ^A T _E X3 syntax	234	\findrootlanguage: removed	62
\@gls@glslink@hyper@update@hook:		footnote: switched to using	
new	120	\glsuppercase	256
\@gls@hypergroup: updated to		\FootnoteNewAcronymDef:	
use etoolbox	272	removed	261, 371
\@gls@noist@toolate: new	189	\glossary: removed	39
\@gls@removedoption: new	27	\glossaryentryfield: removed	232
\@gls@requested@languages:		\glossarymark: switched to using	
new	31	\glsuppercase	52
\@gls@restore@glslocal: new	120	\glossarystyle: removed	235
\@gls@sanitizedesc: removed	24	\glossarysubentryfield:	
\@gls@save@glslocal: new	120	removed	233
\@gls@setupsort@clear: new	18	\gls@automake@makegloss: new	36
\@gls@defaultplural: Removed	80	\gls@automake@option: new	36
\@org@Gls@acentryname: saved		\gls@codepage: set default	
original definition of		codepage to utf8 if	
\@Gls@acentryname	167	\inputencodingname not	
\@xp@gls@getcounterprefix:		defined and switched to	
new	208	\ifdefvoid	33
\@xp@gls@hypernumber: new	237	\gls@delimN: renamed \@delimN	
automake: added makegloss and		to \gls@delimN	237
lite options	36	\gls@delimR: renamed \@delimR	
automakegloss: new	37	to \gls@delimR	237
automakeglosslite: new	37	\gls@disablepagerefexpansion:	
compatible-2.07: deprecated	38	added check for esclocations	202
compatible-3.07: deprecated	38	\gls@noidx@nonumberlist: new	78
\compatibleglossentry:		\gls@noidx@glossary: added	
removed	231, 335	\gls@wrglossdisablelocationcmds	
\compatiblesubglossentry:			199
removed	232, 335	\gls@xdy@set@lang: new	33
\CustomAcronymFields:		\gls@addkey: allow expansion in	
removed	263	PDF bookmarks	89

switched to using			
\glssupercase	90	
\GlsAddXdyAttribute: remove			
duplicate counterless			
attribute definitions	56	
\Glsdisp: new	141	
\glssdisplay: removed	119	
\glssdisplayfirst: removed	...	119	
\glssdoparenifnotempty:			
replaced			
\DeclareRobustCommand with			
\newrobustcmd	261	
\glssdoshowtarget: new	8	
\glssencapwrccontent: new	124	
\Glsentrydesc: allow expansion			
in PDF bookmarks	168	
\Glsentrydescplural: allow			
expansion in PDF			
bookmarks	168	
\Glsentryfirst: allow expansion			
in PDF bookmarks	169	
\Glsentryfirstplural: allow			
expansion in PDF			
bookmarks	170	
\glssentryfmt: obsolete			
commands replaced with			
\glssgenentryfmt	114	
\Glsentrylong: allow expansion			
in PDF bookmarks	173	
\Glsentrylongpl: allow			
expansion in PDF			
bookmarks	173	
\Glsentryname: allow expansion			
in PDF bookmarks	166	
\Glsentryplural: allow			
expansion in PDF			
bookmarks	169	
\Glsentryprefix: allow			
expansion in PDF			
bookmarks	329	
\Glsentryprefixfirst: allow			
expansion in PDF			
bookmarks	328	
\Glsentryprefixfirstplural:			
allow expansion in PDF			
bookmarks	329	
\Glsentryprefixplural: allow			
expansion in PDF			
bookmarks	329	
\Glsentryshort: allow expansion			
in PDF bookmarks	173	
\Glsentryshortpl: allow			
expansion in PDF			
bookmarks	173	
\Glsentrysymbol: allow			
expansion in PDF			
bookmarks	169	
\Glsentrysymbolplural: allow			
expansion in PDF			
bookmarks	169	
\Glsentrytext: allow expansion			
in PDF bookmarks	168	
\glssentrytitlecase: use			
sentence case for PDF			
bookmark	170	
\Glsentryuseri: allow expansion			
in PDF bookmarks	171	
\Glsentryuserii: allow			
expansion in PDF			
bookmarks	171	
\Glsentryuseriii: allow			
expansion in PDF			
bookmarks	171	
\Glsentryuseriv: allow			
expansion in PDF			
bookmarks	172	
\Glsentryuserv: allow expansion			
in PDF bookmarks	172	
\Glsentryuservi: allow			
expansion in PDF			
bookmarks	172	
\glssfielddef: made robust	...	92	
\glssfieldedef: made robust	..	91	
\glssfieldfetch: made robust	..	92	
\glssfieldgdef: made robust	..	91	
\glssfieldxdef: made robust	..	91	
\glssgenacfmt: switched to using			
\glssupercase	116	
\glssgenentryfmt: switched to			
using \glssupercase	115	
\glssgetgrouplabel: converted to			
use \LaTeX 3	234	
\glssgetgrouptitle: made			
robust	234	
\glssglossarymark: switched to			
\glssupercase	51	
\glssifhyper: removed	121	
\glssindexingsetting: new	...	15	
\glssinlineifhaschildren: new		275	

<code>\Glslink</code> : new	122	<code>\SetCustomDisplayStyle</code> :	
<code>\glslocationstoencap</code> : new	50	removed	263
<code>\glslowercase</code> : new	4	<code>\SetCustomStyle</code> : removed	263
<code>\glsmfuaddmap</code> : new	5	<code>\SetDefaultAcronymStyle</code> :	
<code>\glsmfublocker</code> : new	5	removed	260
<code>\glsmfuexcl</code> : new	5	<code>\SetDescriptionAcronymDisplayStyle</code> :	
<code>\glsnavhyperlink</code> : changed		removed	261
<code>\@glo@type</code> to		<code>\SetDescriptionAcronymStyle</code> :	
<code>\currentglossary</code>	270	removed	261
<code>\glsnavhypertarget</code> : changed		<code>\SetDescriptionDUAAcronymDisplayStyle</code> :	
<code>\@glo@type</code> to		removed	261
<code>\currentglossary</code>	271	<code>\SetDescriptionDUAAcronymStyle</code> :	
<code>\glsnavigation</code> : changed		removed	261
<code>\@glo@type</code> to		<code>\SetDescriptionFootnoteAcronymDisplayStyle</code> :	
<code>\currentglossary</code>	273	removed	260
<code>\glsnoidxprenumberlist</code> : new	78	<code>\SetDescriptionFootnoteAcronymStyle</code> :	
<code>\glsnumberlistloop</code> : corrected		removed	261
parameter number	199	<code>\SetDUADisplayStyle</code> : removed	262
<code>\glsseeitemformat</code> : added test		<code>\SetDUASStyle</code> : removed	262
for short field	211	<code>\SetFootnoteAcronymDisplayStyle</code> :	
<code>\glsentencecase</code> : new	4	removed	261
<code>\glsshowtargetfonttext</code> : new	8	<code>\SetFootnoteAcronymStyle</code> :	
<code>\glsshowtargetinner</code> : new	8	removed	261
<code>\glstexpdfstring</code> : new	165	<code>\SetGenericNewAcronym</code> :	
<code>\glsuppercase</code> : new	4	corrected long plural suffix	246
<code>\GlsWarnAddProtectedPageFmt</code> :		switched to using	
new	204	<code>\glsuppercase</code>	246
<code>\glswrglossdisableanchorcmds</code> :		<code>\setglossentrycompatibility</code> :	
new	204	removed	232
<code>\glswrglossdisablelocationcmds</code> :		<code>\SetSmallAcronymDisplayStyle</code> :	
new	203	removed	262
<code>\glswrglosslocationtarget</code> :		<code>\SetSmallAcronymStyle</code> :	
new	204	removed	262
<code>\glswrglosslocationtextfmt</code> :		<code>\SmallNewAcronymDef</code> :	
new	204	removed	262, 371
<code>\ifglsfieldcseq</code> : made robust	94	<code>sort</code> : added clear	15
<code>\ifglsfielddefeq</code> : made robust	93	4.51 – 2022-11-02	
<code>\ifglsfieldeq</code> : made robust	93	General: switch to	
<code>\ifglsfieldvoid</code> : new	68	<code>\TrackIfKnownLanguage</code>	45
<code>\ifglsresetcurrcount</code> : new	108	<code>\do@wrglossary</code> : added	
<code>\ifglsrallowprimitivemods</code> :		<code>\gls@ifnotmeasuring</code>	202
switch default to false	205	<code>\@gls@@automake@immediate</code> :	
<code>\istfile</code> : removed	199	ensure list is expanded before	
<code>kernelglossredefs</code> : corrected		iterating	192
names of choice variables	39	<code>\@gls@if@measuring</code> : new	103
<code>languages</code> : new	31	<code>alttree</code> : replaced <code>\settowidth</code>	
<code>\makeglossary</code> : removed	38	with <code>\glsmeasurewidth</code>	323, 324
<code>mfirstuc</code> : new	41	<code>\gls@if@measuring</code> : new	103
<code>\SetAcronymStyle</code> : removed	262	<code>\gls@local@stepcounter</code> : new	104
		<code>\gls@org@refstepcounter</code> : new	103

<code>\gls@start@measuring:</code> new ..	104	<code>\makeglossaries:</code> ensure list of	
<code>\gls@stop@measuring:</code> new ...	104	glossaries is expanded before	
<code>\glsdohypertarget:</code> added		iterating	195
measuring and group	134	4.52 – 2022-11-03	
<code>\glsfindwidesttoplevelname:</code>		<code>\gls@local@stepcounter:</code>	
replaced <code>\settowidth</code> with		corrected spelling of <code>\@ne</code> .	104
<code>\glsmeasurewidth</code>	322	4.53 – 2023-09-29	
<code>\glsifmeasuring:</code> new	104	<code>\@gls@hypergroup:</code> rewritten in	
<code>\glsmeasuredepth:</code> new	104	L ^A T _E X3	272
<code>\glsmeasureheight:</code> new	104	<code>\glsnavhypergroupdotarget:</code>	
<code>\glsmeasurewidth:</code> new	105	new	272
<code>\glspatchtabularx:</code> switch		<code>\glsnavigationitem:</code> new	273
measuring on and off	105		

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the definition; numbers in *roman* refer to the pages where the entry is used.

Symbols	
<code>\!</code>	<i>128, 130, 187, 448, 449, 697, 699, 700, 730, 951, 953, 984</i>
<code>\"</code>	<i>26, 127, 129, 130, 133, 385, 446–448, 450, 635, 697–700, 702, 889, 950–953, 955, 956</i>
<code>\#</code>	<i>132, 133, 182, 473, 474, 701, 702, 725, 955, 980</i>
<code>\%</code>	<i>178, 184, 471, 475, 576–578, 723, 727, 829–831, 977, 982</i>
<code>\&</code>	<i>45, 175, 396, 470, 646, 722, 902, 976</i>
<code>\'</code>	<i>26, 385, 635, 889</i>
<code>\.</code>	<i>13, 26, 378, 385, 628, 635, 882, 889</i>
<code>\=</code>	<i>26, 385, 635, 889</i>
<code>\?</code>	<i>127, 129, 187, 446–448, 477, 697–699, 729, 950, 952, 984</i>
<code>\@</code>	<i>85, 418, 669, 925</i>
<code>\@delimN</code>	<i>508, 760</i>
<code>\@do@wrglossary</code>	<i>196, 205, 207, <u>208</u>, 483, 489, 490, 735, 740, 741, 990, 996, 997</i>
<code>\@do@esc@wrglossary</code>	<i>204, <u>205</u>, 488, 489, 740, 741, 996</i>
<code>\@do@noesc@wrglossary</code>	<i>204, <u>204</u>, 488, 740, 996</i>
<code>\@do@wrglossary</code>	<i>176, 202, <u>204</u>, 470, 487, 488, 722, 739, 740, 977, 994, 996</i>
<code>\@do@write@glslabels</code>	<i>35, <u>35</u>, 391, 641, 896</i>
<code>\@glo@assign@sortkey</code>	<i>197, 228, 484, 502, 736, 754, 991, 1010</i>
<code>\@glo@list</code>	<i>64, 407, 657, 913</i>
<code>\@glo@sort</code>	<i>26, 385, 635, 889</i>
<code>\@glo@type</code>	<i>213, 492, 744, 1000</i>
<code>\@glossarysec</code>	<i>10, <u>10</u>, 52, 53, 377, 400, 626, 627, 650, 881, 906</i>
<code>\@glossaryseclabel</code>	<i>10, <u>10</u>, 11, 52, 53, 226, 227, 377, 400, 502, 627, 650, 753, 754, 881, 906, 1009</i>
<code>\@glossarysecstar</code>	<i>10, <u>10</u>, 11, 52, 53, 226, 227, 377, 400, 502, 627, 650, 753, 754, 881, 906, 1009</i>
<code>\@gls@default@acr@descplural</code>	<i>260, 774</i>
<code>\@gls@default@acr@firstplural</code>	<i>260, 774</i>
<code>\@gls@default@acr@plural</code>	<i>260, 774</i>
<code>\@gls@checkactual</code>	<i>132, 450, 701, 954, 955</i>
<code>\@gls@checkbar</code>	<i>131, 449, 700, 954</i>
<code>\@gls@checkescactual</code>	<i>129, 447, 448, 698, 699, 952</i>
<code>\@gls@checkescbar</code>	<i>130, 448, 699, 953</i>
<code>\@gls@checkesclevel</code>	<i>130, 131, 448, 449, 699, 700, 953</i>
<code>\@gls@checkescquote</code>	<i>129, 188, 447, 478, 479, 698, 730, 731, 952, 985</i>
<code>\@gls@checklevel</code>	<i>131, 132, 449, 450, 700, 701, 954</i>
<code>\@gls@checkquote</code>	<i>128, 186, 446, 447, 477, 697, 698, 728, 729, 951, 983</i>
<code>\@gls@codepage</code>	<i>190, 191, 986–988</i>
<code>\@gls@default@entryfmt</code>	<i>347, 435, 441, 442, 546, 686, 692, 799</i>
<code>\@gls@expand@field</code>	<i>24, 82, 82, 88, 384, 416, 417, 420, 522–525, 527, 529, 531, 533, 567–571, 634, 667, 670, 671, 774, 776, 777, 779, 781, 782, 784, 786, 821–824, 888, 923, 926, 927</i>
<code>\@gls@extramakeindexopts</code>	<i>186, 193, 476, 482, 728, 734, 983, 989</i>
<code>\@gls@fixbraces</code>	<i>210, <u>210</u>, 491, 743, 998, 999</i>
<code>\@gls@navhypertarget</code>	<i>271, <u>271</u>, 1031</i>
<code>\@gls@noexpand@assign@default@ plurals</code>	<i>260, 774, 821</i>
<code>\@gls@noexpand@field</code>	<i>24, 81, 82, 384, 416, 634, 666, 667, 888, 922, 923</i>
<code>\@gls@noidx@no@sanitizesort</code>	<i>26, 385, 635, 889</i>
<code>\@gls@noidx@nosanitizesort</code>	<i>26, 198, 385, 485, 635, 737, 889, 992</i>

\@gls@noidx@sanitizesort	\@GLSdesc
	25, 385, 634, 889		147, 357, 457, 556, 709, 809, 962
\@gls@nosanitizesort	\@GLSdescplural	.. 148, 458, 709, 963
	25, 25, 198, 384,	\@GLSdescplural@
	385, 485, 634, 737, 888, 889, 992		148, 357, 458, 556, 709, 809, 963
\@gls@sanitizesort	\@GLSfirst 143, 456, 707, 961
	25, 25, 198, 384,	\@GLSfirst@
	385, 485, 634, 737, 888, 889, 992		143, 356, 456, 555, 707, 808, 961
\@gls@xdycheckbackslash	\@GLSfirstplural	. 145, 456, 708, 962
	133, 134, 451, 702, 703, 956	\@GLSfirstplural@
\@gls@xdycheckquote		145, 356, 457, 555, 708, 809, 962
	133, 450, 702, 955, 956	\@GLSname 146, 457, 708, 962
\@gls@localreset 106, 106,	\@GLSname@
	109, 431, 433, 682, 683, 939, 940		146, 357, 457, 556, 709, 809, 962
\@gls@localunset 106, 106,	\@GLSpl 140, 140, 454, 706, 959
	109, 431, 432, 682, 683, 939, 940	\@GLSpl@
\@gls@reset 106, 107,		140, 334, 454, 594, 706, 847, 959
	109, 431, 432, 682, 683, 939, 940	\@GLSplural 144, 456, 708, 961
\@gls@showtarget 7,	\@GLSplural@
	8, 9, 375, 376, 625, 626, 879, 880		144, 356, 456, 555, 708, 808, 961
\@gls@unset 106, 106,	\@GLSsymbol 149, 458, 710, 963
	109, 431, 432, 682, 683, 939, 940	\@GLSsymbol@
\@newglossaryentry@defcounters		149, 358, 458, 556, 710, 809, 963
	108, 108, 432, 683, 940	\@GLSsymbolplural	150, 458, 710, 964
\@this@glo@ 64, 407, 657, 913	\@GLSsymbolplural@
\@ACRfull	... 243, 244, 510, 762, 1017		150, 358, 458, 556, 710, 810, 964
\@ACRfullpl	.. 244, 510, 511, 762, 1017	\@GLStext 142, 455, 707, 960
\@ACRlong 162, 242, 355, 464,	\@GLStext@
	510, 554, 715, 762, 807, 969, 1016		142, 355, 455, 555, 707, 808, 960
\@ACRlongpl 164,	\@GLSuseri 151, 459, 710, 964
	243, 465, 511, 716, 763, 970, 1016	\@GLSuseri@
\@ACRshort	... 158, 242, 354, 461,		151, 358, 459, 557, 710, 810, 964
	510, 553, 713, 762, 807, 967, 1016	\@GLSuserii 152, 459, 711, 964
\@ACRshortpl 160, 243,	\@GLSuserii@
	462, 463, 511, 714, 763, 968, 1016		152, 359, 459, 557, 711, 810, 964
\@Acrfull 243, 510, 762, 1017	\@GLSuseriii 153, 459, 711, 965
\@Acrfullpl 244, 510, 762, 1017	\@GLSuseriii@
\@Acrlong 161, 242, 354, 463,		153, 359, 459, 557, 711, 810, 965
	510, 554, 715, 762, 807, 969, 1016	\@GLSuseriv 154, 460, 711, 965
\@Acrlongpl 163,	\@GLSuseriv@
	243, 464, 510, 716, 762, 970, 1016		154, 359, 460, 557, 711, 810, 965
\@Acrshort	\@GLSuserv 155, 460, 712, 966
	157, 353, 461, 553, 713, 806, 967	\@GLSuserv@
\@Acrshortpl 159, 462, 714, 968		155, 360, 460, 558, 712, 811, 966
\@Alph 203, 206,	\@GLSuservi 156, 460, 712, 966
	207, 488, 489, 740, 741, 995–997	\@GLSuservi@
\@GLS 137, 453, 704, 705, 958		156, 360, 460, 558, 712, 811, 966
\@GLS@ 137,	\@Gls 136, 137, 452, 704, 958
	137, 333, 453, 593, 705, 846, 958		
\@GLSdesc 147, 457, 709, 962		

<code>\@Gls@</code>	109, 111, 137, 137, 332, 433, 434, 452, 592, 684, 685, 704, 845, 941, 942, 958	<code>\@Glsuseri</code>	151, 459, 710, 964
<code>\@Gls@link</code>	122, 947	<code>\@Glsuseri@</code>	151, 358, 459, 710, 964
<code>\@Gls@acentryname</code>	166, 245, 466, 511, 717, 763, 972, 1018	<code>\@Glsuserii</code>	152, 459, 711, 964
<code>\@Gls@entry@field</code> 165, 165, 166, 167, 421, 465–469, 671, 717–720, 971, 972	<code>\@Glsuserii@</code>	152, 359, 459, 711, 964
<code>\@Gls@entryname</code>	166, 166, 245, 466, 511, 717, 763, 971, 972, 1018	<code>\@Glsuseriii</code>	153, 459, 711, 965
<code>\@GlsSetXdyFirstLetterAfterDigits</code>	178, 179, 471, 723, 977, 978	<code>\@Glsuseriii@</code>	153, 359, 459, 711, 965
<code>\@GlsSetXdyNumberGroupOrder</code>	179, 471, 472, 723, 978	<code>\@Glsuseriv</code>	154, 460, 711, 965
<code>\@Glsdesc</code>	147, 457, 709, 962	<code>\@Glsuseriv@</code>	154, 359, 460, 711, 965
<code>\@Glsdesc@</code>	147, 357, 457, 556, 709, 809, 962	<code>\@Glsuserv</code>	155, 460, 712, 965
<code>\@Glsdescplural</code>	148, 457, 709, 963	<code>\@Glsuserv@</code>	155, 360, 460, 712, 965
<code>\@Glsdescplural@</code>	148, 357, 457, 556, 709, 809, 963	<code>\@Glsuservi</code>	156, 460, 712, 966
<code>\@Glsdisp</code>	141, 141, 960	<code>\@Glsuservi@</code>	156, 360, 460, 712, 966
<code>\@Glsentryfield</code>	89, 165, 168–173, 328, 329, 927, 971–975	<code>\@Mi</code>	290, 603, 857
<code>\@Glsfirst</code>	143, 456, 707, 961	<code>\@PGLS</code>	333, 333, 593, 846
<code>\@Glsfirst@</code>	143, 356, 456, 555, 707, 808, 961	<code>\@PGLS@</code>	333, 333, 593, 846
<code>\@Glsfirstplural</code>	145, 456, 708, 961	<code>\@PGLSpl</code>	333, 334, 593, 846
<code>\@Glsfirstplural@</code>	145, 356, 456, 708, 961	<code>\@PGLSpl@</code>	334, 334, 593, 846
<code>\@Glsname</code>	146, 457, 708, 962	<code>\@Pgls</code>	331, 331, 592, 845
<code>\@Glsname@</code>	146, 357, 457, 555, 708, 809, 962	<code>\@Pgls@</code>	331, 331, 592, 845
<code>\@Glspl</code>	139, 454, 705, 959	<code>\@Pglspl</code>	332, 332, 592, 845
<code>\@Glspl@</code>	110, 113, 139, 139, 333, 433, 435, 454, 592, 593, 684, 685, 705, 846, 941, 943, 959	<code>\@Pglspl@</code>	332, 332, 592, 845
<code>\@Glsplural</code>	144, 456, 708, 961	<code>\@Roman</code>	203, 206, 207, 488, 489, 740, 741, 995, 996
<code>\@Glsplural@</code>	144, 356, 456, 555, 708, 808, 961	<code>\@acrfull</code>	243, 243, 509, 761, 1016
<code>\@Glsymbol</code>	149, 458, 709, 963	<code>\@acrfullpl</code>	244, 510, 762, 1017
<code>\@Glsymbol@</code>	149, 358, 458, 556, 709, 809, 963	<code>\@acrlong</code>	160, 242, 354, 463, 509, 554, 714, 715, 761, 807, 968, 969, 1016
<code>\@Glsymbolplural</code> 150, 458, 710, 963		<code>\@acrlongpl</code>	162, 242, 464, 510, 716, 762, 970, 1016
<code>\@Glsymbolplural@</code>	150, 358, 458, 556, 710, 810, 963	<code>\@acrshort</code>	157, 242, 353, 461, 509, 510, 553, 712, 761, 762, 806, 966, 1016
<code>\@Glstext</code>	142, 455, 707, 960	<code>\@acrshortpl</code>	158, 159, 242, 243, 462, 510, 713, 762, 967, 1016
<code>\@Glstext@</code>	142, 355, 455, 555, 707, 808, 960	<code>\@addtoacronymlists</code> 21, 382, 632, 886	
<code>\@Glsuser@i</code> 358–360, 557, 558, 810, 811		<code>\@addtoacronymlists</code>	21
		<code>\@afterheading</code> 278, 580, 597, 833, 851	
		<code>\@alph</code>	203, 206, 207, 488, 489, 740, 741, 995–997
		<code>\@arabic</code>	203, 206, 207, 488, 489, 740, 741, 995, 996
		<code>\@auxout</code>	71, 73, 110, 193, 194, 196, 199, 211, 216, 272, 411, 434, 482–484, 486, 492, 494, 594, 661, 662, 684, 685, 734–736, 738, 744, 746, 847, 917, 918, 941, 942, 989, 991, 993, 999, 1001, 1002, 1031

<code>\backslashchar</code> .. 127, 128, 133, 134, 187, 445, 451, 696, 697, 702, 703, 730, 950, 951, 956, 984	<code>\do@seeglossary</code> 196, 209, 210, 483, 491, 735, 742, 743, 991, 998, 999
<code>\bsphack</code> 201, 487, 739, 994	<code>\do@subglossentry</code> 505, 538, 757, 791
<code>\cGls</code> 111, 111, 434, 685, 942	<code>\do@wrglossary</code> .. 124, 202, 444, 487, 574, 694, 739, 827, 948, 994
<code>\cGls@</code> 109, 111, 111, 433, 434, 684, 685, 941, 942	<code>\do@write@glslabels</code> 35, 36, 391, 641, 895, 896
<code>\cGlspl</code> 112, 434, 685, 942	<code>\do@writeaux@info</code> 211, 492, 744, 999
<code>\cGlspl@</code> 110, 112, 112, 433–435, 684, 685, 941, 942	<code>\domakeglossaries</code> ... 34, 34, 35, 193, 195, 390, 391, 482, 483, 640, 641, 734, 735, 895, 989, 990
<code>\cclv</code> 290, 603, 604, 857	<code>\ehc</code> 290, 603, 857
<code>\cglS</code> 111, 111, 434, 685, 942	<code>\empty</code> 14, 18, 21, 38, 40, 41, 43, 55, 56, 60, 62, 63, 97, 102, 125, 136–140, 157–164, 180, 183, 185, 190–192, 200, 201, 209, 235, 353–355, 379, 381, 382, 392, 394, 395, 401, 402, 404, 406, 407, 426, 429, 444, 452–454, 461–465, 472, 474, 476, 480, 481, 486, 487, 490, 506, 522, 524–531, 533, 535, 553, 554, 574, 576–578, 628, 630, 632, 642, 644, 645, 651, 652, 655, 657, 676, 677, 680, 695, 704–706, 712–717, 724, 726, 728, 731–733, 738, 739, 742, 758, 774, 776, 778, 780, 782–784, 786, 788, 806–808, 828, 830, 831, 882, 884–886, 897–901, 907, 908, 910, 912, 913, 932, 933, 936, 948, 949, 957–959, 966–971, 978, 981, 982, 986–988, 993, 994, 998, 1013
<code>\cglS@</code> 109, 111, 111, 433, 434, 683, 685, 940, 942	<code>\end@fixbraces</code> 210, 491, 743, 998, 999
<code>\cglSpl</code> 112, 112, 434, 685, 942	<code>\endfortrue</code> 31, 67, 87, 389, 409, 419, 594, 639, 659, 670, 847, 893, 914, 926, 1031
<code>\cglSpl@</code> 109, 112, 112, 433, 434, 684, 685, 941, 942	<code>\esphack</code> 202, 487, 739, 994
<code>\chapter</code> 43, 395, 645, 901	<code>\expandtwoargs</code> ... 41, 394, 644, 900
<code>\classoptionslist</code> . 41, 394, 644, 900	<code>\@firstofone</code> 26, 35, 277, 385, 386, 391, 635, 641, 850, 889, 890, 895
<code>\closegls</code> 190, 190, 479, 480, 731, 732, 986, 987	<code>\@firstofthree</code> 121, 135, 136, 138, 141, 157, 159, 161, 163, 353– 355, 442, 452, 453, 455, 461– 464, 553, 554, 693, 704–706, 712, 714–716, 806–808, 947, 957, 959, 960, 966, 967, 969, 970
<code>\colht</code> 290, 604, 857	<code>\@firstoftwo</code> 29–31, 86, 87, 104, 121, 138–140, 159, 160, 163, 164,
<code>\colroom</code> 290, 291, 604, 857	
<code>\currentlabelname</code> 11, 227, 377, 502, 627, 754, 881, 1009	
<code>\curroptions</code> 41, 394, 644, 900	
<code>\declaredoptions</code> . 41, 394, 644, 900	
<code>\delimN</code> 508, 760	
<code>\delimR</code> 508, 760	
<code>\disable@onlypremakeg</code> 43, 194, 395, 482, 646, 734, 901, 989	
<code>\disable@premakecs</code> 43, 43, 396, 646, 901	
<code>\disabled@glSaddxdycounters</code> 55, 57, 402, 652, 653, 908	
<code>\do@addcounter</code> 55, 401, 402, 652, 907	
<code>\do@auxoutstuff</code> 216, 494, 746, 1001, 1002	
<code>\do@glossentry</code> .. 504, 538, 756, 791	
<code>\do@glS@checkesc</code> 128, 697, 951	
<code>\do@glS@getcounterprefix</code> 208, 488, 489, 740, 741, 997, 998	
<code>\do@glS@islistofacronyms</code> 21, 382, 632, 886	
<code>\do@glSsee</code> 100, 428, 679, 935	
<code>\do@ifinlist</code> 55, 401, 651, 907	
<code>\do@newglossaryentry</code> 246, 511, 521–534, 567– 571, 763, 773–788, 820–824, 1018	

388, 389, 418, 419, 442, 453,	736, 748–750, 883, 991, 1004, 1005
454, 462–465, 637–639, 669,	\@glo@defaultcounter
670, 693, 705, 706, 713, 714, 98, 99, 427, 678, 934
716, 717, 892, 893, 925, 926,	\@glo@desc 75, 94,
937, 947, 959, 967, 968, 970, 971	95, 97, 99, 413, 424–428, 663,
\@for 31, 41,	675, 676, 678, 919, 931, 932, 934
43, 45, 55, 56, 63, 64, 86, 87,	\@glo@descaccess
127, 180–182, 192, 194, 195,	337, 339, 340, 539–541, 792–794
202, 210, 217, 247, 389, 394,	\@glo@descplural 75, 94, 95, 413,
395, 401, 402, 407, 418, 419,	424, 425, 663, 675, 919, 931, 932
445, 472, 473, 482, 483, 488,	\@glo@descpluralaccess
491, 495, 512, 522, 524, 526,	337, 339, 340, 539–541, 792–794
527, 529, 532, 533, 535, 574,	\@glo@do@sortentries
594, 595, 639, 644, 646, 652,	217, <u>217</u> , 495, 746, 747, 1002, 1003
657, 669, 670, 696, 724, 725,	\@glo@entry 177, 471, 722, 977
734, 735, 740, 743, 746, 747,	\@glo@entryprefix
764, 774, 777, 778, 780, 782, 327, 328, 589, 590, 842, 843
785, 786, 788, 828, 847, 848,	\@glo@entryprefixfirst
893, 900–902, 907, 908, 913, 327, 328, 589, 590, 842, 843
925, 926, 950, 978–980, 988–	\@glo@entryprefixfirstplural . .
990, 995, 999, 1002, 1019, 1031 327, 328, 589, 590, 842, 843
\@glo@@desc 99, 427, 678, 934	\@glo@entryprefixplural
\@glo@@symbol 99, 428, 679, 935 327, 328, 589, 590, 842, 843
\@glo@access . . 336, 339, 344, 538,	\@glo@esclabel 102,
540, 541, 543, 544, 792–794, 797	103, 429, 430, 680, 681, 936, 937
\@glo@addchildren 217, <u>217</u> ,	\@glo@etext 435–437, 686–688
221, 495, 498, 747, 750, <u>1002</u> , 1005	\@glo@first . . . 76, 95, 98, 99, 413,
\@glo@assign@sortkey	425, 427, 428, 531, 570, 663,
. . . . 194, 197, 228, 483, 484,	676, 678, 784, 824, 919, 932, 934
502, 734, 736, 754, 990, 991, 1010	\@glo@firstaccess
\@glo@autosee . 100, <u>100</u> , 428, 679, 935 336, 339, 539–541, 792–794
\@glo@autoseehook	\@glo@firstplural
. 100, <u>101</u> , 428, 679, 935	76, 95, 98, 413, 425, 427, 570,
\@glo@check@mkidxrangear	663, 676, 678, 824, 919, 932, 934
. 126, 208, 445,	\@glo@firstpluralaccess
490, 574, 696, 741, 827, 949, 997 336, 339, 539–541, 792–794
\@glo@check@sortallowed	\@glo@grabfirst
. 16, <u>16</u> , 17–19, 195, 223, <u>223</u> , 499, 751, 1006
198, 379–381, 483, 485, 629–	\@glo@label 78,
631, 735, 737, 883–885, 990, 992	80, 88, 89, 91–100, 108, 175,
\@glo@childlist . 217, 495, 747, 1002	176, 322, 328, 339, 340, 415,
\@glo@counter 77, 95, 99, 414, 425,	420, 422–428, 432, 469, 470,
427, 664, 676, 678, 920, 932, 934	540–542, 590, 621, 622, 665,
\@glo@counterprefix . . . 199, 204,	666, 671, 673–679, 683, 721,
207–209, 235, 238, 486, 488–	722, 794, 795, 843, 875, 920–
490, 506, 508, 738, 740–742,	922, 927, 929–935, 940, 976, 977
758, 760, 993, 996–998, 1013, 1015	\@glo@list 100, 428, 679, 935
\@glo@default@sorttype	\@glo@long 68, 80,
. 15, <u>15</u> , 197, 219–	96, 99, 409, 415, 425, 427, 659,
221, 379, 484, 497, 498, 629,	665, 676, 678, 915, 921, 932, 934

<code>\@glo@longaccess</code>	678, 773–775, 777, 779, 781, 784, 785, 820, 824, 921, 932, 934
<code>\@glo@longpl</code>	80, 96, 99, 415, 425, 427, 521, 523, 525, 527, 528, 531, 532, 567, 665, 676, 678, 773, 774, 776, 777, 779, 781, 784, 785, 820, 921, 932, 934
<code>\@glo@longpluralaccess</code>	337, 339, 340, 539–541, 792–794
<code>\@glo@name</code>	16, 75, 81, 95, 97–99, 380, 413, 416, 425– 427, 629, 663, 666, 675, 677, 678, 883, 919, 922, 931, 933, 934
<code>\@glo@no@assign@sortkey</code> ..	194, 228, 483, 502, 734, 754, 990, 1010
<code>\@glo@nonumberlist</code>	79, 414, 415, 665, 921
<code>\@glo@numfmt</code>	208, 490, 574, 742, 827, 997
<code>\@glo@parent</code>	18, 78, 95, 97, 98, 102, 103, 218, 381, 414, 425, 426, 429, 430, 495, 630, 664, 676, 677, 680, 681, 747, 884, 920, 932, 933, 936, 937, 1003
<code>\@glo@plural</code>	76, 95, 97, 98, 413, 425–427, 569, 663, 676–678, 822, 919, 932–934
<code>\@glo@pluralaccess</code>	336, 339, 539–541, 792–794
<code>\@glo@prefix</code>	12, 78, 96, 102, 103, 126, 208, 378, 414, 425, 429, 430, 445, 490, 574, 627, 664, 676, 680, 681, 696, 742, 827, 882, 920, 932, 936, 937, 949, 950, 997
<code>\@glo@range</code>	208, 490, 574, 742, 827, 997
<code>\@glo@see</code> ..	77, 95, 100, 414, 425, 428, 664, 676, 679, 920, 932, 935
<code>\@glo@seeautonumberlist</code> ...	12, 12, 77, 378, 414, 627, 664, 881, 920
<code>\@glo@short</code> ...	68, 80, 96, 99, 409, 415, 425, 427, 570, 660, 665, 676, 678, 824, 915, 921, 932, 934
<code>\@glo@shortaccess</code>	337, 339, 340, 539–541, 567–570, 792–794, 820–822, 824
<code>\@glo@shortpl</code>	80, 96, 99, 415, 425, 427, 521, 523, 525, 527, 528, 531, 532, 567, 570, 665, 676,
<code>\@glo@shortpluralaccess</code>	337, 339, 340, 539–541, 792–794
<code>\@glo@sort</code>	16, 18, 25, 26, 76, 95, 98, 102, 103, 380, 381, 385, 413, 425, 427, 429, 430, 629, 631, 634, 635, 663, 676, 677, 680, 681, 883– 885, 889, 919, 932, 933, 936, 937
<code>\@glo@sortedinsert</code>	218, 218, 495, 496, 747, 748, 1003
<code>\@glo@sortentries</code>	216, 219–221, 494, 497, 498, 746, 748, 749, 1002, 1004, 1005
<code>\@glo@sorthandler@case</code> ...	219, 220, 496, 497, 748, 749, 1003, 1005
<code>\@glo@sorthandler@letter</code> ..	219, 220, 496, 497, 748, 749, 1003, 1004
<code>\@glo@sorthandler@nocase</code> ..	219, 221, 496, 498, 748, 749, 1004, 1005
<code>\@glo@sorthandler@word</code>	218, 219, 496, 497, 748, 1003, 1004
<code>\@glo@sortinghandler</code>	216, 218, 494, 496, 746, 748, 1002, 1003
<code>\@glo@sortinglist</code>	216–218, 221, 494–496, 498, 746, 747, 750, 1002, 1003, 1005
<code>\@glo@sortmacro@case</code>	220, 497, 749, 1005
<code>\@glo@sortmacro@def</code>	221, 498, 750, 1005
<code>\@glo@sortmacro@def@do</code>	221, 498, 750, 1005
<code>\@glo@sortmacro@letter</code>	220, 497, 749, 1004
<code>\@glo@sortmacro@nocase</code>	220, 497, 749, 1005
<code>\@glo@sortmacro@standard</code>	220, 497, 749, 1004
<code>\@glo@sortmacro@use</code>	221, 498, 750, 1006
<code>\@glo@sortmacro@word</code>	219, 496, 748, 1004
<code>\@glo@sorttype</code>	197, 222, 223, 228, 484, 498, 499, 502, 736, 750, 751, 754, 991, 1006, 1010
<code>\@glo@storeentry</code>	16–18, 102, 379–381, 429, 629, 630, 680, 883–885, 936

\@glo@suffix . . 126, 208, 445, 490,
574, 696, 742, 827, 949, 950, 997
\@glo@symbol 68,
76, 95, 99, 100, 409, 413, 425,
428, 526, 530, 659, 663, 676,
679, 779, 783, 915, 919, 932, 935
\@glo@symbolaccess 336,
339, 539–541, 570, 792–794, 823
\@glo@symbolplural
. 77, 95, 100, 413, 425,
428, 663, 676, 679, 919, 932, 935
\@glo@symbolpluralaccess
336, 339, 340, 539–541, 792–794
\@glo@text
76, 95, 98, 99, 136–141, 165–
167, 413, 425, 427, 428, 452–
455, 465, 466, 527, 569, 590,
663, 676, 677, 679, 704–706,
717, 718, 779, 822, 843, 919,
932, 933, 935, 958–960, 971, 972
\@glo@textaccess 336, 339, 538, 540,
541, 567–569, 792–794, 820–823
\@glo@thislabel 101, 429, 679, 680, 936
\@glo@thislettergrp 222–
224, 499, 500, 751, 752, 1006, 1007
\@glo@thisvalue 69, 70, 410, 660, 916
\@glo@tmp 71, 88, 89, 209,
420, 490, 671, 742, 917, 927, 998
\@glo@type 11,
18, 77, 95–100, 176, 177, 194,
200, 213–218, 221, 222, 226,
227, 245, 377, 381, 413, 425–
428, 470, 471, 482, 486, 487,
493–495, 498, 499, 501, 502,
511, 522, 524, 525, 527, 529,
531, 533, 535, 594, 595, 627,
630, 663, 675–679, 722, 734,
738, 739, 744–747, 750, 751,
753, 754, 763, 774, 776, 778,
780, 782, 784, 786, 788, 847,
848, 881, 884, 885, 919, 932–
935, 977, 989, 993, 994, 1000–
1003, 1005, 1006, 1008, 1009, 1018
\@glo@types 35, 63, 64, 70, 72,
107, 177, 192, 194, 195, 268,
322, 391, 407, 411, 431, 432,
471, 482, 483, 537, 621, 641,
657, 661, 682, 683, 722, 734,
735, 790, 875, 896, 913, 916,
917, 939, 940, 977, 988–990, 1030
\@glo@useri . . 79, 96, 99, 415, 425,
427, 665, 676, 678, 921, 932, 934
\@glo@useriaccess
337, 339, 340, 539–541, 792–794
\@glo@userii . 79, 96, 99, 415, 425,
427, 665, 676, 678, 921, 932, 934
\@glo@useriiaccess 337,
339, 340, 539–541, 792, 793, 795
\@glo@useriii 79, 96, 99, 415, 425,
427, 665, 676, 678, 921, 932, 934
\@glo@useriiiaccess 337,
339, 340, 539–541, 792, 793, 795
\@glo@useriv . 79, 96, 99, 415, 425,
427, 665, 676, 678, 921, 932, 934
\@glo@userivaccess
. . . . 338–340, 539–541, 793–795
\@glo@userv . . 79, 96, 99, 415, 425,
427, 665, 676, 678, 921, 932, 934
\@glo@uservaccess
338–340, 539, 540, 542, 793–795
\@glo@uservi . 80, 96, 99, 415, 425,
427, 665, 676, 678, 921, 932, 934
\@glo@userviaccess
338–340, 539, 540, 542, 793–795
\@glodesc 99, 428, 678, 935
\@glolist@ 97, 426, 677, 933
\@gloname 99, 427, 678, 934
\@glossary@default@style
. 11, 11, 13, 213,
235, 263, 377, 378, 493, 507,
535, 627, 628, 745, 759, 788,
881, 882, 1000, 1013, 1028, 1029
\@glossaryentryfield 101,
102, 103, 429, 430, 680, 936, 937
\@glossarysection
. . 51, 52, 399, 400, 649, 650, 905
\@glossarystyle 213, 214, 226, 493,
501, 745, 753, 1000, 1001, 1009
\@glossarysubentryfield 101, 102,
103, 429, 430, 680, 681, 936, 937
\@gls 136, 136, 452, 704, 957
\@gls@ 109, 111, 136, 136, 330, 332,
433, 434, 452, 591, 592, 684,
685, 704, 844, 845, 941, 942, 957
\@gls@@automake@immediate
192, 193, 482, 733, 734, 988, 989
\@gls@@link 122, 122, 442, 443, 693, 947
\@gls@Hcounter
. . . . 124, 125, 444, 695, 948, 949

<code>\@gls@ReturnAfterFi</code>	<code>\@gls@checklevel</code>
..... 238, 508, 509, 760, 1015	128, <u>131</u> , 187, 446, 449, 478,
<code>\@gls@access@display</code> .. <u>344</u> , 543, 797	697, 700, 701, 729, 951, 954, 984
<code>\@gls@actualchar</code>	<code>\@gls@checkmkidxchars</code>
..... 103, 129, <u>132</u> , 184, 430, 102, 126, <u>127</u> , 186, 196,
447, 448, 450, 475, 577, 680,	207, 209, 429, 445, 446, 477,
681, 698, 699, 701, 726, 727,	483, 489, 491, 574, 680, 696,
830, 936, 937, 952, 954, 955, 981	697, 729, 735, 741, 742, 827,
<code>\@gls@addpredefinedattributes</code> .	936, 949, 950, 984, 990, 997, 998
. <u>57</u> , 180, 189, 403, 472, 479,	<code>\@gls@checkquote</code>
653, 724, 731, 908, 909, 978, 986	127, <u>128</u> ,
<code>\@gls@adjustmode</code>	186, 446, 447, 476, 477, 697,
..... 176, <u>177</u> , 470, 722, 977	698, 728, 729, 950, 951, 983, 984
<code>\@gls@after</code> ... 21, 382, 383, 632, 887	<code>\@gls@classI</code>
<code>\@gls@assign@symbol@field</code> 180, 181, 472, 473, 724, 979
<u>25</u>	<code>\@gls@classII</code> 181, 473, 724, 979
<code>\@gls@assign@symbolplural@field</code> <u>25</u>	<code>\@gls@codepage</code>
<code>\@gls@automake</code> <u>190</u> , 195, 479, 480,	62,
483, 731, 732, 735, 986, 987, 990	216, 406, 494, 656, 746, 912, 1002
<code>\@gls@automake@immediate</code> .. <u>191</u> ,	<code>\@gls@counter</code>
192, 480–482, 732–734, 987, 988	119,
<code>\@gls@automake@types</code>	123, 125, 176, 199, 208, 209,
..... <u>192</u> , <u>192</u> , 195, 988, 990	442–445, 470, 486, 490, 574,
<code>\@gls@before</code>	693–695, 722, 738, 742, 827,
..... 21, 382, 632, 887	946, 948, 949, 977, 993, 997, 998
<code>\@gls@between</code>	<code>\@gls@counterwithin</code>
... 273, 594, 595, 848, 1031, 1032	.. 14, <u>15</u> , 379, 628, 629, 882, 883
<code>\@gls@body</code> ... 167, 466, 717, 718, 972	<code>\@gls@ctr</code>
<code>\@gls@checkactual</code>	55, 401, 652, 907
127, <u>132</u> , 187, 446, 450, 477,	<code>\@gls@currentlettergroup</code> .. 222,
697, 701, 729, 951, 954, 955, 984	224, 498, 500, 750, 752, 1006, 1007
<code>\@gls@checkboxar</code>	<code>\@gls@debugfalse</code> ... 7, 375, 625, 879
127, <u>131</u> , 187, 446, 449, 478,	<code>\@gls@debugtrue</code>
697, 700, 729, 951, 953, 954, 984 7, 375, 624, 625, 878, 879
<code>\@gls@checkedmkidx</code> 126–134, 186–	<code>\@gls@declareoption</code>
188, 445–451, 476–478, 696– <u>11</u> , 12, 13, 20, 23, 24,
703, 728–730, 950–956, 983–985	30, 33, 34, 36, 37, 40, 41, 377,
<code>\@gls@checkesc</code>	378, 381–384, 388, 390, 391,
<u>128</u> , <u>132</u> , 187, 697, 730, 951, 984	393, 394, 627, 628, 631–634,
<code>\@gls@checkescactual</code>	638, 640, 641, 643, 644, 881,
..... 127, <u>129</u> , 187, 446–448,	882, 885–888, 892, 894–897, 899
477, 697–699, 729, 950, 952, 984	<code>\@gls@default</code> 113, 435, 686, 943
<code>\@gls@checkescbar</code>	<code>\@gls@default@value</code>
..... 127, <u>130</u> , 187, 446, 448, 68–70, <u>76</u> , 81, 82,
478, 697, 699, 729, 951–953, 984	95, 97–99, 262, 328, 409, 410,
<code>\@gls@checkesclevel</code>	413, 416, 417, 425–428, 530,
..... 128, <u>130</u> , 187, 448, 449,	589, 590, 659, 660, 663, 666,
697, 699, 700, 730, 951, 953, 984	667, 675–678, 783, 843, 915,
<code>\@gls@checkescquote</code> 127, <u>129</u> , 187,	916, 919, 922, 923, 932–935, 1028
188, 446, 447, 477–479, 697,	<code>\@gls@defaultttype</code>
698, 729, 730, 950–952, 984, 985	71, <u>71</u> , 917
	<code>\@gls@deffile</code>
	84–
	86, 418, 419, 668, 669, 924, 925
	<code>\@gls@define@glossaryentrycounter</code>
	... 14, <u>14</u> , 42, 227, 229, 379,

395, 502, 503, 628, 629, 645, 754, 755, 882, 883, 900, 1009, 1010	<code>\@gls@dothiswrite</code> 190– 192, 480, 481, 732, 733, 986–988
<code>\@gls@define@glossarysubentrycounter</code> 15, 42, 227, 229, 379, 395, 502, 503, 629, 645, 754, 755, 883, 901, 1009, 1010	<code>\@gls@elem</code> 594, 847, 1031
<code>\@gls@defsort</code> 16–19, 99, 380, 381, 428, 629–631, 679, 883–885, 935	<code>\@gls@enablesavenonumberlist</code> 79, 85, 414, 418, 665, 669, 921, 925
<code>\@gls@defsortcount</code> 16–19, 73, 380, 381, 412, 629–631, 662, 883–885, 918	<code>\@gls@encapchar</code> 130, 131, 184, 208, 210, 448, 449, 475, 490, 491, 574, 577, 699, 700, 726, 727, 742, 743, 827, 830, 953, 954, 981, 997, 998
<code>\@gls@deprecated@acr@warning</code> 262, 776, 778, 780, 782, 784, 786, 1028	<code>\@gls@entry@count</code> 110, 111, 433, 434, 684, 685, 941, 942
<code>\@gls@do@acronymsdef</code> 20, 20, 42, 74, 381, 382, 395, 412, 631, 632, 645, 662, 886, 900, 901, 918	<code>\@gls@entry@field</code> 88, 89, 108, 164, 165, 166, 168–174, 340–342, 420, 421, 432, 465– 469, 542, 543, 671, 683, 717– 721, 795, 796, 927, 940, 971–975
<code>\@gls@do@automake@err</code> 36, 37, 38, 896, 897	<code>\@gls@escbsdq</code> 126, 127, 185, 445, 446, 476, 578, 696, 697, 727, 728, 831, 950, 982
<code>\@gls@do@glsunset</code> 120, 120, 136–141, 946, 958–960	<code>\@gls@expand@fields</code> 82, 82, 83, 417, 667, 923
<code>\@gls@do@indexdef</code> 40–43, 74, 394, 395, 412, 644, 645, 662, 899–901, 918	<code>\@gls@expandonce</code> . . 82, 417, 667, 923
<code>\@gls@do@numbersdef</code> 40, 42, 74, 393–395, 412, 644, 645, 662, 899–901, 918	<code>\@gls@extramakeindexopts</code> . . 188, 193, 479, 482, 731, 734, 985, 989
<code>\@gls@do@symbolsdef</code> 40, 74, 393, 412, 643, 662, 899, 918	<code>\@gls@fetchfield</code> 69, 87, 410, 419, 660, 670, 916, 926
<code>\@gls@do@symbolssdef</code> 42, 395, 645, 900, 901	<code>\@gls@field@link</code> 89, 90, 141, 142–156, 355–360, 421, 422, 455–461, 555–558, 672, 707–712, 808–811, 928, 960–966
<code>\@gls@doautomake</code> 37, 37, 38, 193, 195, 392, 482, 483, 642, 734, 735, 897, 989, 990	<code>\@gls@fieldaccess@display</code> 344, 345–347, 544–546, 797–799
<code>\@gls@docheckquotedef</code> 186– 188, 476–479, 728–731, 983–985	<code>\@gls@firsttok</code> 222, 223, 499, 751, 1006
<code>\@gls@docloadedfalse</code> 6, 374, 624, 878	<code>\@gls@fixbraces</code> 100, 210, 428, 491, 679, 743, 935, 998
<code>\@gls@docloadedtrue</code> . 6, 374, 624, 878	<code>\@gls@forbidtexext</code> 71, 73, 411, 661, 662, 917
<code>\@gls@dodflistparser</code> 194, 483, 734, 989, 990	<code>\@gls@fulllist</code> 35, 641, 896
<code>\@gls@doentrycounterdef</code> 42, 43, 395, 645, 900, 901	<code>\@gls@get@counterprefix</code> 209, 490, 742, 998
<code>\@gls@doentrydef</code> . 441, 442, 692, 693	<code>\@gls@getbody</code> 167, 466, 717, 972
<code>\@gls@dolast</code> . . 210, 211, 491, 743, 999	<code>\@gls@getcounter</code> . . 74, 412, 662, 918
<code>\@gls@donext</code> . . 210, 211, 491, 743, 999	<code>\@gls@getcounterprefix</code> 208, 209, 488–490, 740–742, 998
<code>\@gls@donext@def</code> 175, 470, 721, 722, 976	<code>\@gls@getgrouptitle</code> 196, 234, 234, 273, 483, 506, 595, 735, 758, 848, 991, 1012, 1032
<code>\@gls@dosubentrycounterdef</code> 42, 43, 395, 645, 901	<code>\@gls@glossary</code> 201, 201, 487, 739, 994

<code>\@gls@glslink@hyper@update@hook</code>	<code>\@gls@link@checkfirsthyper</code>
. 120, 120 , 946 122 , 136–141, 443, 452–
<code>\@gls@gobbleopt</code> . . . 72, 411, 661, 917	455, 694, 704–706, 947, 957–960
<code>\@gls@grptitle</code>	<code>\@gls@link@label</code> 123,
234, 270, 273, 506, 594, 595,	443, 523, 528, 694, 775, 781, 948
758, 847, 848, 1012, 1031, 1032	<code>\@gls@link@nocheckfirsthyper</code> . .
<code>\@gls@hyp@opt</code> 89,	141, 141 , 157–164, 455, 461–
90, 111, 112, 121 , 122, 136–	465, 707, 712–716, 960, 966–971
164, 243, 244, 330–333, 421,	<code>\@gls@link@opts</code> 123,
422, 434, 442, 452–465, 509,	443, 523, 528, 694, 775, 781, 948
510, 591–593, 672, 685, 693,	<code>\@gls@list</code> . . 594, 595, 847, 848, 1031
704–716, 761, 762, 844–846,	<code>\@gls@listsuffix</code> . . 55, 401, 651, 907
928, 942, 947, 957–970, 1016, 1017	<code>\@gls@loadlist</code> 13, 13 ,
<code>\@gls@hyp@opt@cs</code> . 121, 442, 693, 947	263, 378, 535, 628, 788, 882, 1028
<code>\@gls@hypergroup</code>	<code>\@gls@loadlong</code> . . 12, 12 , 13, 263,
. 272, 272 , 594, 847, 1031	378, 535, 627, 628, 788, 882, 1028
<code>\@gls@if@measuring</code> 103, 103 , 937	<code>\@gls@loadsuper</code> 12, 12 , 13,
<code>\@gls@ifinlist</code>	263, 378, 535, 628, 788, 882, 1028
. 55, 55 , 401, 651, 652, 907	<code>\@gls@loadtree</code> 13, 13 ,
<code>\@gls@ifnotmeasuring</code> 430, 681	263, 378, 535, 628, 788, 882, 1028
<code>\@gls@igtype</code> 75, 413, 663, 919	<code>\@gls@local@increment@currcount</code>
<code>\@gls@increment@currcount</code> . 109, 109,
110 , 432, 433, 683, 684, 940, 941	110 , 432, 433, 683, 684, 940, 941
<code>\@gls@indexdef</code> . 40, 41 , 394, 644, 899	<code>\@gls@loclist</code> 197, 198,
<code>\@gls@initnonumberlist</code>	223, 224, 484, 485, 499, 500,
. 78 , 79, 79 , 96, 414,	736, 737, 751, 752, 991, 992, 1007
425, 664, 665, 676, 920, 921, 932	<code>\@gls@map</code> 86,
<code>\@gls@islistofacronyms</code>	87, 418, 419, 669, 670, 925, 926
. 21, 382, 632, 886	<code>\@gls@missinglang@warn</code> 23,
<code>\@gls@keylist</code> 371, 567, 820	23 , 46, 384, 397, 633, 647, 888, 903
<code>\@gls@keymap</code>	<code>\@gls@missingnumberlist</code> . . . 81 ,
. 79, 86, 86 , 87–89, 328, 338,	99, 416, 427, 666, 678, 922, 934
415, 418–420, 540, 589, 665,	<code>\@gls@noaccess</code>
669–671, 793, 842, 921, 925–927	338, 344, 540, 543, 544, 793, 797
<code>\@gls@label</code> 196, 199, 205, 207, 208,	<code>\@gls@noexpand@field</code> 81
483, 484, 486, 489, 490, 735,	<code>\@gls@noexpand@fields</code> 81 ,
738, 740–742, 991, 993, 996, 997	83, 416, 417, 666, 667, 922, 923
<code>\@gls@labelsfile</code> . . 35, 391, 641, 896	<code>\@gls@nohyperlist</code> 22,
<code>\@gls@langmod</code> 33, 190, 191, 216, 480,	22 , 74, 75, 123, 383, 412, 413,
481, 732, 733, 894, 986, 987, 1001	443, 633, 663, 694, 887, 919, 948
<code>\@gls@levelchar</code>	<code>\@gls@noidx@do</code>
103, 130, 131, 184, 430, 448,	222, 223 , 499, 750, 751, 1006, 1007
449, 475, 577, 681, 699, 700,	<code>\@gls@noidx@getgrouptitle</code> . 196,
726, 727, 830, 937, 953, 954, 981	234 , 483, 506, 735, 758, 991, 1012
<code>\@gls@link</code> . . . 122, 123 , 136–141,	<code>\@gls@noidx@sanitizesort</code> . . . 25,
157–164, 353–355, 443, 452–	198, 385, 485, 634, 737, 889, 992
455, 461–465, 553, 554, 693,	<code>\@gls@noidx@setsanitizesort</code> . . .
694, 704–707, 712–717, 806– 28, 28 , 198,
808, 947, 948, 958–960, 966–971	387, 485, 636, 637, 737, 891, 992

<code>\@gls@noidxloclist@finalsep</code> . . .	475–478, 577, 698–701, 726–730, 830, 951–955, 981, 983, 985
. 197, 198, 485, 736, 737, 992	
<code>\@gls@noidxloclist@prev</code>	<code>\@gls@reference</code> 196,
. 198, 224, 225, 485,	199, 225, 483, 484, 486, 501,
500, 501, 737, 752, 753, 992, 1008	735, 736, 738, 753, 991, 993, 1008
<code>\@gls@noidxloclist@sep</code>	<code>\@gls@removedoption</code>
. 197, 224, 225, 484, 485, 27, 31, 32, 38, 890, 894, 898
500, 501, 736, 752, 753, 992, 1008	<code>\@gls@removespaces</code>
<code>\@gls@noist@toolate</code> 238, 508, 760, 1014, 1015
. 184, 185, 189, 981, 982, 986	<code>\@gls@renewglossary</code> 189,
<code>\@gls@noref@warn</code>	201, 479, 487, 731, 739, 986, 994
. 196, 199, 222, 483, 486,	<code>\@gls@replacementtext</code>
499, 735, 738, 751, 990, 993, 1006 343, 344, 543, 796, 797
<code>\@gls@notranslatorhook</code> 29	<code>\@gls@requested@languages</code>
<code>\@gls@numberlink</code> 31, 31, 45, 893, 902
. 237, 238, 508, 760, 1014	<code>\@gls@rest</code> 167, 466, 717, 718, 972
<code>\@gls@numbersdef</code>	<code>\@gls@restore@glslocal</code>
. 40, 40, 393, 394, 644, 899 120, 124, 946, 948
<code>\@gls@numlist@lastsep</code>	<code>\@gls@restreat</code> 85, 418, 669, 925
. 175, 470, 721, 722, 976	<code>\@gls@roman</code> 58,
<code>\@gls@numlist@nextsep</code>	403, 404, 575, 654, 828, 909, 910
. 175, 470, 721, 722, 976	<code>\@gls@sanitized@tmp</code>
<code>\@gls@numlist@sep</code> 127, 445, 446, 696, 950
. 175, 470, 721, 722, 976	<code>\@gls@sanitizedesc</code>
<code>\@gls@old@chapter</code> 43, 395, 645, 901 24, 384, 389, 634, 639
<code>\@gls@oldnewglossaryentryposthook</code>	<code>\@gls@sanitizename</code> 25, 384, 634, 888
. 339, 540, 794	<code>\@gls@sanitizesort</code> 16,
<code>\@gls@oldnewglossaryentryprehook</code>	25, 380, 384, 629, 634, 883, 888
. 338, 339, 540, 793	<code>\@gls@sanitizesymbol</code>
<code>\@gls@onlypremakeg</code> 25, 384, 389, 390, 634, 639, 888
. 43, 43, 395, 645, 646, 901	<code>\@gls@save@glslocal</code>
<code>\@gls@order</code> 190– 120, 120, 124, 946, 948
192, 480, 481, 731–733, 986–988	<code>\@gls@saveentrycounter</code>
<code>\@gls@org@LT@output</code> 290, 603, 857 124, 124, 176,
<code>\@gls@org@glsnoidxdisplayloc</code> . .	444, 470, 694, 695, 722, 948, 977
. 198, 485, 737, 992	<code>\@gls@savenonumberlist</code> 78, 78, 79,
<code>\@gls@org@glsseeformat</code>	198, 414, 664, 665, 920, 921, 993
. 198, 485, 737, 992	<code>\@gls@see@noindex</code> 9, 9, 78,
<code>\@gls@override@glossary</code> 39,	376, 377, 414, 626, 664, 880, 920
39, 40, 392, 393, 643, 898, 899	<code>\@gls@setacrstyle</code> 31, 42, 389, 390,
<code>\@gls@patchtabularx</code> 105, 430, 681, 938	394, 395, 639, 645, 893, 900, 901
<code>\@gls@preglossaryhook</code> 213,	<code>\@gls@setcounter</code> 73, 74, 412, 662, 918
215, 492, 493, 744, 745, 1000, 1001	<code>\@gls@setdefault@glslink@opts</code> .
<code>\@gls@prevlevel</code> 301, 302, 323– 123, 124, 443, 694, 948
325, 584, 585, 610, 611, 622,	<code>\@gls@setsort</code> 16–19, 124,
623, 837, 838, 863, 864, 876, 877	176, 380, 381, 444, 470, 629–
<code>\@gls@provide@newglossary</code>	631, 694, 722, 884, 885, 948, 977
. 71, 73, 411, 661, 662, 916–918	<code>\@gls@setupshortcuts</code>
<code>\@gls@quotechar</code> 128– 42, 394, 395, 645, 900, 901
132, 184–186, 188, 446–450,	<code>\@gls@setupsort@clear</code> 18, 885

780, 782, 784–786, 788, 790,
 886, 887, 913, 1018, 1019, 1030
 \@glsaddkey 88, 89, 420, 671, 927
 \@glsaddstoragekey
 . . 87, 88, 420, 670, 671, 926, 927
 \@glsaddxdyattribute 56, 402, 652, 908
 \@glsdefaultplural 80, 416, 666
 \@glsdefaultsort 16,
 81, 380, 416, 629, 666, 883, 922
 \@glsdesc 146, 147, 457, 709, 962
 \@glsdesc@
 147, 357, 457, 556, 709, 809, 962
 \@glsdescplural
 147, 148, 457, 709, 962, 963
 \@glsdescplural@
 148, 357, 457, 556, 709, 809, 963
 \@glsdisp 140, 140, 141, 455, 706, 960
 \@glsentry
 . 35, 107, 110, 391, 431, 432,
 434, 641, 682–685, 896, 939–942
 \@glsentrytitlecase
 170, 170, 467, 468, 719, 974
 \@glsfirst . . . 143, 456, 707, 960, 961
 \@glsfirst@
 143, 356, 456, 555, 707, 808, 961
 \@glsfirstletter
 178, 178, 179, 471, 723, 977, 978
 \@glsfirstplural
 144, 145, 456, 708, 961
 \@glsfirstplural@ 145,
 356, 356, 456, 555, 708, 808, 961
 \@glsnumber
 . . . 237, 237, 508, 759, 760, 1014
 \@glsisacronymlistfalse
 22, 383, 632, 887
 \@glsisacronymlisttrue
 22, 383, 632, 887
 \@glslink 104, 124, 134,
 135, 176, 270, 444, 451, 452,
 470, 594, 695, 703, 704, 722,
 847, 937, 938, 948, 957, 977, 1031
 \@glslocalreset 105, 106,
 109, 431, 432, 681–683, 938–940
 \@glslocalunset 106, 106,
 109, 431, 432, 682, 683, 939, 940
 \@glslocref 199, 204, 205,
 207, 208, 486, 488–490, 574,
 738, 740–742, 827, 993, 996, 997
 \@glsminrange 179,
 179, 180, 181, 472, 473, 575,
 576, 723–725, 828, 829, 978, 979
 \@glsname 145, 146, 457, 708, 962
 \@glsname@
 146, 357, 457, 555, 708, 809, 962
 \@glsnavhypertarget
 272, 272, 594, 847, 1031
 \@glsnextpages 214,
 228, 493, 503, 745, 754, 1001, 1010
 \@glsnodesc 80, 95,
 97, 99, 415, 425, 428, 666, 675,
 676, 678, 922, 931, 932, 934, 935
 \@glsnoname 80,
 95, 97, 99, 415, 425–427, 665,
 675, 677, 678, 921, 931, 933, 934
 \@glsnonextpages 214,
 228, 493, 502, 745, 754, 1001, 1010
 \@glsnumberformat
 120, 123, 176, 199,
 208, 442, 443, 470, 486, 490,
 574, 693, 694, 722, 738, 741,
 742, 827, 946, 948, 977, 993, 997
 \@glsopenfile 189, 190,
 200, 479, 487, 731, 738, 986, 994
 \@glsorder 32, 193,
 194, 390, 482, 639, 734, 894, 989
 \@glspl 138, 453, 705, 958
 \@glspl@ 109, 112, 138, 138, 331–333,
 433, 434, 453, 591–593, 684,
 685, 705, 845, 846, 941, 942, 958
 \@glsplural . . . 143, 144, 456, 708, 961
 \@glsplural@
 144, 356, 456, 555, 708, 808, 961
 \@glsprefix@record@hook . . 330,
 330, 331–334, 591–593, 844–847
 \@glsreset 105, 106,
 109, 430–432, 681–683, 938–940
 \@glssee 100,
 210, 428, 491, 679, 743, 935, 999
 \@glsshowaccsupp . 7, 9, 336, 375,
 376, 538, 625, 626, 791, 879, 880
 \@glsshowtarget 7, 8, 8, 375,
 376, 451, 624, 625, 703, 878–880
 \@glsymbol . . . 148, 149, 458, 709, 963
 \@glsymbol@
 149, 358, 458, 556, 709, 809, 963
 \@glsymbolplural
 149, 150, 458, 710, 963

<code>\@glsymbolplural@</code>	<code>\@ifglossaryexists</code> ..
150, 358 , 458, 556, 710, 809, 963	64, 64 , 212,
<code>\@glstarget</code>	407, 492, 657, 658, 744, 913, 999
135,	<code>\@ifl@t@r</code>
135 , 231, 272, 452, 504, 594,	27, 386, 636, 890
703, 704, 756, 847, 957, 1011, 1031	<code>\@ifnextchar</code>
<code>\@glstext</code>	73,
142, 455, 707, 960	121, 412, 442, 662, 693, 918, 947
<code>\@glstext@</code>	<code>\@ifpackageloaded</code>
142, 355 , 455, 555, 707, 808, 960	6, 11, 12,
<code>\@glsunset</code>	29–31, 46, 62, 103, 174, 186,
106, 106 ,	334, 374, 377, 388, 389, 396,
109, 431, 432, 682, 683, 939, 940	406, 430, 469, 476, 537, 624,
<code>\@glsuseri</code> ...	627, 637–639, 646, 657, 681,
150, 151, 458, 710, 964	721, 728, 790, 878, 881, 882,
<code>\@glsuseri@</code>	892, 893, 902, 912, 937, 976, 983
151, 358 , 458, 557, 710, 810, 964	<code>\@ifstar</code>
<code>\@glsuserii</code> ...	64, 72, 87, 88,
151, 152, 459, 710, 964	121, 178, 179, 240, 407, 411,
<code>\@glsuserii@</code>	420, 442, 471, 509, 657, 661,
152, 359 , 459, 557, 710, 810, 964	670, 671, 693, 723, 761, 913,
<code>\@glsuseriii</code>	917, 926, 927, 947, 977, 978, 1015
....	<code>\@ifundefined</code>
152, 153, 459, 711, 964, 965	45,
<code>\@glsuseriii@</code>	280, 291, 303, 310, 324, 396,
153, 359 , 459, 557, 711, 810, 965	585, 594, 598, 604, 611, 615,
<code>\@glsuseriv</code> ...	622, 623, 646, 838, 847, 852,
153, 154, 459, 711, 965	858, 864, 865, 868, 869, 876, 902
<code>\@glsuseriv@</code>	<code>\@ignored@glossaries</code>
154, 359 , 460, 557, 711, 810, 965
<code>\@glsuserv</code> ...	35, 74, 75, 75 , 391, 412,
154, 155, 460, 711, 965	413, 641, 662, 663, 896, 918, 919
<code>\@glsuserv@</code>	<code>\@input@</code>
155, 360 , 460, 557, 712, 811, 965	215, 494, 745, 1001
<code>\@glsuservi</code> ...	<code>\@istfilename</code>
155, 156, 460, 712, 966	48, 193,
<code>\@glsuservi@</code>	194, 398, 482, 648, 734, 904, 989
156, 360 , 460, 558, 712, 811, 966	<code>\@makecol</code>
<code>\@glswidestname</code> 322, 322 , 323, 324,	290, 603, 604, 857
584, 585, 622, 623, 838, 875, 876	<code>\@makeglossaries@warn@noprintglossary</code>
<code>\@glswritefiles</code>
38,	194, 195 , 734, 735, 989, 990
200 , 392, 486, 642, 738, 897, 993	<code>\@makeglossary</code>
<code>\@glsxtr@doaccsupp</code>	34,
335, 537, 791	189 , 194, 390, 391, 479, 482,
<code>\@glsxtr@record</code>	640, 641, 731, 734, 895, 986, 989
330, 591, 844	<code>\@minus</code>
<code>\@gobble</code>	276, 297,
7, 16–18, 34–36, 86, 104, 127,	316, 597, 607, 618, 850, 861, 872
178, 182, 194, 196, 276, 375,	<code>\@mkboth</code> ..
380, 381, 390, 391, 418, 419,	52, 399, 400, 649, 650, 905
430, 445, 471, 473, 474, 482,	<code>\@one</code>
483, 573, 576, 577, 624, 625,	104, 937
629–631, 640, 641, 669, 670,	<code>\@newglossary</code>
681, 696, 722, 723, 725, 734,	71,
735, 826, 829, 830, 850, 878,	73 , 73 , 411, 412, 661, 662, 917, 918
879, 884, 885, 895, 896, 925,	<code>\@newglossaryentry@defcounters</code>
926, 937, 950, 977, 980, 989, 990
<code>\@gobblethree</code>	100, 101 , 108,
7, 375, 625, 879	428, 429, 432, 679, 683, 935, 940
<code>\@idxitem</code>	<code>\@newglossaryentryposthook</code>
317, 618, 872
<code>\@ifclassloaded</code> 6, 14, 51, 374, 378,	88, 89, 100, 101 ,
399, 624, 628, 649, 878, 882, 905	328, 339, 420, 428, 429, 540,
	590, 671, 679, 794, 843, 927, 935
	<code>\@newglossaryentryprehook</code>

	88, 89, 94, 96, 101 ,
	328, 338, 339, 420, 424, 425,

429, 540, 589, 671, 675, 676,
 679, 793, 843, 927, 931, 932, 935
 \@nil 21, 100, 126–128,
 167, 186, 187, 208, 210, 222–
 224, 237, 238, 383, 428, 445,
 446, 466, 477, 478, 490, 491,
 499, 500, 508, 574, 632, 679,
 696, 697, 717, 729, 730, 741,
 743, 751, 752, 759, 760, 827,
 887, 935, 949–951, 972, 984,
 997, 998, 1006, 1007, 1014, 1015
 \@nnil 21,
 210, 383, 491, 632, 743, 887, 999
 \@no@makeglossaries
 34, 194, 197, 199,
 390, 391, 483, 484, 486, 640,
 734, 736, 738, 895, 990, 991, 993
 \@no@post@desc
 47, 398, 579, 648, 832, 903
 \@nopostdesc 47,
 214, 397, 493, 648, 745, 903, 1001
 \@onelevel@sanitize 8, 25,
 35, 58, 86, 102, 126, 127, 178,
 179, 183, 209, 211, 223, 238,
 376, 385, 404, 418, 430, 445,
 446, 471, 472, 474, 491, 492,
 499, 575, 576, 625, 634, 635,
 641, 654, 669, 680, 696, 723,
 726, 742, 744, 751, 828, 830,
 880, 889, 896, 909, 925, 936,
 950, 978, 980, 998, 999, 1006, 1015
 \@onlypreamble . . . 73, 83, 94, 95,
 110, 113, 195, 198, 412, 418,
 424, 433, 435, 483, 485, 662,
 668, 675, 684, 686, 735, 737,
 918, 924, 931, 941, 943, 990, 993
 \@onlypremakeg 43,
 48, 49, 55, 57, 60, 73, 185, 395,
 398, 402, 405, 412, 476, 645,
 648, 652, 653, 655, 662, 728,
 901, 904, 907, 908, 911, 918, 983
 \@org@Gls@acrentryname . . . 167, 972
 \@org@glossaryentrynumbers
 . . . 214, 215, 493, 745, 1000, 1001
 \@org@Gls@assign@descplural
 522, 529,
 531, 533, 567, 570, 571, 774,
 781, 782, 784, 786, 821, 823, 824
 \@org@Gls@assign@firstpl
 522–525, 527,
 529, 531–533, 567–571, 773,
 774, 776–782, 784, 786, 820–824
 \@org@Gls@assign@plural 522–525,
 527, 529, 531–533, 567–571,
 774, 776–782, 784, 786, 821–824
 \@org@Gls@assign@symbolplural
 522–525, 527,
 531–533, 568, 569, 571, 774,
 776–780, 784, 786, 821, 822, 824
 \@org@Gls@numberformat
 175, 469, 721, 976
 \@org@newglossaryentryprehook
 94, 424, 675, 931
 \@outputpage 290, 603, 604, 857
 \@p@glossarysection 51,
 53, 399, 400, 649, 650, 905, 906
 \@p@Gls 330, 330, 591, 844
 \@p@Gls@ 330, 330, 591, 844
 \@p@Gls@pl 331, 331, 591, 844
 \@p@Gls@pl@ 331, 331, 591, 844
 \@plus 276, 297,
 316, 597, 607, 618, 850, 861, 872
 \@print@glossary 212,
 215, 492, 494, 744, 745, 1000, 1001
 \@print@noidx@glossary . . . 213,
 221, 492, 498, 744, 750, 1000, 1006
 \@print@gloss@checkexists
 212, 212,
 213, 214, 492, 493, 744, 745, 1000
 \@print@gloss@checkexists@allowignored
 212, 492, 744, 1000
 \@print@gloss@checkexists@noignored
 212, 212, 213, 492, 744, 999, 1000
 \@print@gloss@setsort . . 194, 197,
 213, 214, 483, 484, 492, 493,
 734, 736, 744, 745, 990, 991, 1000
 \@print@glossary
 . . . 212, 213, 213, 492, 744, 1000
 \@roman . . . 58, 403, 575, 654, 828, 909
 \@secondofthree 121, 135,
 137, 139, 157, 159, 161, 163,
 353, 442, 452–454, 461–463,
 465, 553, 693, 704, 706, 713–
 716, 806, 947, 957–959, 967–970
 \@secondoftwo 26,
 29–31, 46, 86, 87, 104, 135–
 138, 141, 157, 158, 161, 162,
 204, 276, 277, 326, 353–355,
 385, 386, 388, 389, 396, 418,
 419, 452, 453, 455, 461, 463,

464, 553, 554, 572, 589, 635, 637–639, 646, 669, 670, 703– 706, 712, 713, 715, 806–808, 825, 842, 850, 889, 890, 892, 893, 902, 925, 926, 937, 938, 957, 958, 960, 966, 967, 969, 996	\@set@glo@numformat 126, 208, 445, 490, 574, 696, 742, 827, 949, 997	\@sglsaddkey 88, 420, 671, 927	\@sglsaddstoragekey 87, 420, 670, 926	\@tabacckludge 26, 385, 635, 889	\@text@composite@x . 26, 385, 635, 889	\@thirdofthree . . . 121, 138, 140, 158, 160, 162, 164, 354, 442, 453, 454, 461, 463–465, 553, 693, 705, 706, 713–715, 717, 807, 947, 958, 959, 967–969, 971	\@this@attr . . . 182, 473, 474, 725, 980	\@this@childlabel 217, 495, 747, 1002	\@this@counter 56, 402, 652, 908	\@this@ctr 182, 473, 725, 980	\@this@key 87, 419, 670, 926	\@this@label . 217, 495, 746, 747, 1002	\@this@cs 43, 395, 396, 646, 901	\@tmp 404, 575, 828	\@use@ption 41, 394, 644, 900	\@warn@nomakeglossaries . . 192, 216, 482, 494, 733, 746, 988, 1002	\@wrglossary@pageformat 203, 204, 488, 740, 995	\@wrglossarynumberhook 203, 204, 207, 488, 489, 740, 741, 995, 997	\@write@glslabelsname 35, 36, 641, 896	\@xdy@main@language 33, 33, 390, 480, 481, 494, 640, 732, 733, 746, 894	\@xdy@attributelist 54, 56, 182, 401, 402, 473, 651, 652, 725, 907, 908, 980	\@xdy@attributes 54, 56, 180, 401, 402, 472, 573, 575, 651, 652, 724, 826, 828, 907, 908, 979	\@xdy@counters 55, 56, 182, 401, 402, 473, 651, 652, 725, 907, 908, 980	\@xdy@crossrefhook 59, 181, 404, 473, 655, 725, 910, 980	\@xdy@language 62, 216, 406, 494, 656, 746, 912, 1001	\@xdy@lettergroups 63, 63, 183, 407, 475, 577, 657, 726, 830, 912, 913, 981	\@xdy@locationclassorder 60, 61, 181, 405, 473, 576, 655, 725, 829, 911, 980	\@xdy@locref 55, 56, 183, 401, 402, 474, 573, 577, 651, 652, 726, 826, 830, 907, 908, 981	\@xdy@numbergrouporder 63, 179, 179, 407, 471, 472, 657, 723, 913, 978	\@xdy@requiredstyles 61, 61, 180, 406, 472, 574, 656, 724, 828, 912, 978	\@xdy@sortrules . . 61, 61, 183, 405, 475, 577, 656, 726, 830, 911, 981	\@xdy@style 180, 472, 574, 575, 724, 828, 978, 979	\@xdy@useralphabets 57, 58, 180, 403, 472, 575, 653, 724, 828, 909, 979	\@xdy@userlocationdefs 59, 60, 181, 404, 405, 473, 573, 576, 654, 655, 725, 827, 829, 910, 911, 979	\@xdy@userlocationnames 59, 60, 404, 405, 574, 654, 655, 827, 910, 911	\@xfor@nextelement 210, 491, 743, 999	\@xp@gl@getcounterprefix 205, 207, 208, 996, 997	\@xp@glshypernumber . . 237, 237, 1014	\\ 101, 127, 132, 178, 184, 185, 237, 238, 429, 445, 471, 475, 476, 508, 577, 578, 580, 581, 587, 588, 680, 696, 701, 702, 723, 727, 760, 830, 831, 833, 834, 840, 841, 936, 950, 955, 977, 981, 982, 1014, 1015	\^ 26, 35, 385, 635, 641, 889, 896	\^ 26, 385, 635, 889	\ 127, 130, 132, 187, 446, 448, 478, 697, 699, 701, 729, 951–953, 955, 984	\~ 26, 386, 635, 889	A	\A 234, 1012	\a 26, 386, 635, 890	\AA 27, 386, 635, 890	\aa 27, 386, 635, 890
--	---	--	---------------------------------------	--	--	--	---	---------------------------------------	--	---	--	---	--	-------------------------------	---	--	--	---	--	---	--	---	--	---	--	---	--	---	--	--	--	---	--	---	--	---------------------------------------	---	--	--	--	--------------------------------	---	--------------------------------	----------	------------------------	--------------------------------	---------------------------------	---------------------------------

<code>\Ac</code>	259 , 521 , 773 , 1028	<code>\Acrfullplfmt</code> 243 , 244 , 246 , 254 , 256 ,
<code>\ac</code>	259 , 521 , 773 , 1028		367 , 369 , 510 , 512 , 518 , 519 ,
<code>access (key)</code>	336		563 , 565 , 762 , 764 , 770 , 771 ,
<code>\accsuppglossaryentryfield</code>	361 , 538 , 559 , 791 , 812		816 , 818 , 1016–1018 , 1025 , 1026
<code>\accsuppglossarysubentryfield</code>	361 , 538 , 559 , 791 , 812	<code>\acrfullplfmt</code> 242 , 244 , 246 , 254 , 256 ,
<code>\Acf</code>	259 , 521 , 773 , 1028		366 , 369 , 510 , 512 , 518 , 519 ,
<code>\acf</code>	259 , 521 , 773 , 1028		563 , 565 , 762 , 764 , 770 , 771 ,
<code>\Acfp</code>	259 , 521 , 773 , 1028		816 , 818 , 1016–1018 , 1024 , 1026
<code>\acfp</code>	259 , 521 , 773 , 1028	<code>\acrlinkfootnote</code> 260 , 260 , 522 , 775 , 1028
<code>\Acl</code>	259 , 521 , 773 , 1028	<code>\acrlinkfullformat</code>	242 ,
<code>\acl</code>	258 , 521 , 773 , 1028		242 , 243 , 509–511 , 761–763 , 1016
<code>\Aclp</code>	259 , 521 , 773 , 1028	<code>\ACRlong</code>	162 , 464 , 715 , 969
<code>\aclp</code>	258 , 521 , 773 , 1028	<code>\Acrlong</code>	161 , 161 ,
<code>\Acp</code>	259 , 521 , 773 , 1028		259 , 463 , 521 , 715 , 773 , 969 , 1028
<code>\acp</code>	259 , 521 , 773 , 1028	<code>\acrlong</code>	160 ,
<code>\acrfootnote</code>	260 , 522 , 523 , 528 , 775 , 781 , 1028		258 , 463 , 521 , 714 , 773 , 968 , 1028
<code>\ACRfull</code>	243 , 510 , 762 , 1017	<code>\ACRlongpl</code>	164 , 465 , 716 , 970
<code>\Acrfull</code>	243 ,	<code>\Acrlongpl</code>	163 , 163 ,
	259 , 510 , 521 , 761 , 773 , 1017 , 1028		259 , 464 , 521 , 716 , 773 , 970 , 1028
<code>\acrfull</code>	243 , 243 , 259 , 509 ,	<code>\acrlongpl</code>	162 ,
	521 , 761 , 773 , 1016 , 1017 , 1028		258 , 464 , 521 , 716 , 773 , 970 , 1028
<code>\ACRfullfmt</code>	242 , 244 , 246 , 254 , 256 ,	<code>\acrnameformat</code>	245 ,
	366 , 368 , 510 , 512 , 518 , 519 ,		511 , 526 , 569 , 763 , 779 , 822 , 1017
	563 , 565 , 762 , 763 , 770 , 771 ,	<code>\acrnohreffootnote</code>	260 , 522 , 775 , 1028
	816 , 818 , 1016–1018 , 1024 , 1026	<code>acronym (option)</code>	20
<code>\Acrfullfmt</code>	242 , 243 , 246 , 254 , 256 ,	<code>\acronymentry</code>
	366 , 368 , 510 , 511 , 518 , 519 ,	 246 , 246 , 248–253 , 255–258 ,
	563 , 565 , 762 , 763 , 770 , 771 ,		362–365 , 367–371 , 511–521 ,
	816 , 818 , 1016–1018 , 1024 , 1026		559–566 , 763–773 , 813–815 ,
<code>\acrfullfmt</code>	242 , 243 , 246 , 254 , 256 ,		817–820 , 1018–1023 , 1025–1027
	366 , 368 , 509 , 511 , 518 , 519 ,	<code>\acronymfont</code>	116 , 117 ,
	563 , 564 , 761 , 763 , 769 , 771 ,		157–160 , 167 , 174 , 245 , 245 ,
	816 , 818 , 1016–1018 , 1024 , 1026		247–258 , 350 , 351 , 353–355 ,
<code>\acrfullformat</code>	174 ,		362–371 , 439 , 440 , 461–463 ,
	242 , 242 , 469 , 509 , 510 , 521 ,		466 , 469 , 511–521 , 523 , 524 ,
	534 , 721 , 761 , 773 , 787 , 975 , 1016		526–532 , 551–554 , 559–566 ,
<code>\ACRfullpl</code>	244 , 510 , 762 , 1017		568–570 , 690 , 691 , 712–714 ,
<code>\Acrfullpl</code>	244 ,		718 , 721 , 763–773 , 775 , 777 ,
	259 , 510 , 521 , 762 , 773 , 1017 , 1028		778 , 780–783 , 785 , 804–808 ,
<code>\acrfullpl</code>	244 , 244 ,		813–821 , 823 , 944 , 945 , 966–
	259 , 510 , 521 , 762 , 773 , 1017 , 1028		968 , 972 , 975 , 1017 , 1019–1027
<code>\ACRfullplfmt</code> 243 , 244 , 246 , 255 , 256 ,	<code>acronymlists (option)</code>	22
	367 , 369 , 511 , 512 , 518 , 519 ,	<code>\acronymname</code>
	563 , 565 , 762–764 , 770 , 771 ,	 20 , 44 , 47 , 381 , 382 , 396 , 397 ,
	816 , 818 , 1016–1018 , 1025 , 1026		631 , 632 , 646 , 647 , 886 , 901 , 903
		<code>acronyms (option)</code>	20

`\acronymsort` . . . [246](#), [247](#), [248–253](#),
[255–258](#), [362–365](#), [367](#), [368](#),
[370](#), [371](#), [511–521](#), [559–561](#),
[564](#), [566](#), [763–773](#), [813–815](#),
[817–820](#), [1018–1023](#), [1025–1027](#)
`\acronymtype` [19](#), [20](#), [245–247](#), [381](#),
[382](#), [511](#), [512](#), [521–535](#), [567–](#)
[570](#), [631](#), [632](#), [763](#), [764](#), [773–](#)
[788](#), [820–823](#), [885](#), [886](#), [1018](#), [1019](#)
`\acrpluralsuffix` [241](#),
[246](#), [248–251](#), [255–257](#), [362](#),
[363](#), [367–370](#), [509](#), [511](#), [513–](#)
[515](#), [518–525](#), [527–529](#), [531](#),
[532](#), [534](#), [559](#), [560](#), [564](#), [566–](#)
[571](#), [761](#), [763](#), [765–767](#), [770–](#)
[773](#), [775–777](#), [779–782](#), [784](#),
[785](#), [787](#), [788](#), [813](#), [817–824](#),
[1016](#), [1018](#), [1020–1022](#), [1025–1027](#)
`\ACRshort` [158](#), [461](#), [713](#), [967](#)
`\Acrshort` [157](#), [157](#),
[258](#), [461](#), [521](#), [713](#), [773](#), [966](#), [1027](#)
`\acrshort` [156](#),
[258](#), [461](#), [521](#), [712](#), [773](#), [966](#), [1027](#)
`\ACRshortpl` [160](#), [462](#), [714](#), [968](#)
`\Acrshortpl` [159](#), [159](#),
[258](#), [462](#), [521](#), [714](#), [773](#), [968](#), [1027](#)
`\acrshortpl` [158](#),
[258](#), [462](#), [521](#), [713](#), [773](#), [967](#), [1027](#)
acrstyles:
`dua` [253](#), [365](#)
`dua-desc` [255](#), [367](#)
`footnote` [255](#), [368](#)
`footnote-desc` [257](#), [370](#)
`footnote-sc` [256](#), [369](#)
`footnote-sc-desc` [257](#), [370](#)
`footnote-sm` [257](#), [370](#)
`footnote-sm-desc` [258](#), [370](#)
`long-sc-short` [250](#)
`long-sc-short-desc` [251](#), [363](#)
`long-short` [248](#), [362](#)
`long-short-desc` [251](#), [363](#)
`long-sm-short` [250](#)
`long-sm-short-desc` [252](#), [364](#)
`long-sp-short` [249](#)
`long-sp-short-desc` [251](#)
`sc-short-long` [250](#)
`sc-short-long-desc` [252](#), [364](#)
`short-long` [249](#), [362](#)
`short-long-desc` [252](#), [364](#)
`sm-short-long` [251](#)
`sm-short-long-desc` [253](#), [365](#)
`\Acs` [258](#), [521](#), [773](#), [1027](#)
`\acs` [258](#), [521](#), [773](#), [1027](#)
`\Acsp` [258](#), [521](#), [773](#), [1027](#)
`\acsp` [258](#), [521](#), [773](#), [1027](#)
`\add@accent@` [26](#), [385](#), [635](#), [889](#)
`\addcontentsline` [54](#), [401](#), [651](#), [907](#)
`\addglossarytocaptions`
. [45](#), [46](#), [396](#), [646](#), [647](#), [902](#)
`\addtolength` [324](#), [585](#), [623](#), [838](#), [876](#)
`\advance`
. [17](#), [18](#), [97](#), [104](#), [125](#), [290](#), [380](#),
[381](#), [426](#), [444](#), [603](#), [630](#), [677](#),
[695](#), [857](#), [884](#), [885](#), [933](#), [937](#), [949](#)
`\AE` [26](#), [386](#), [635](#), [890](#)
`\ae` [26](#), [386](#), [635](#), [890](#)
`altlist` (style) [278](#)
`altlistgroup` (style) [278](#)
`altlisthypergroup` (style) [279](#)
`altlong4col` (style) [285](#)
`altlong4col-booktabs` (style) [288](#)
`altlong4colborder` (style) [286](#)
`altlong4colheader` (style) [285](#)
`altlong4colheaderborder` (style) [286](#)
`altlongragged4col` (style) [295](#)
`altlongragged4col-booktabs`
(style) [289](#)
`altlongragged4colborder` (style) [296](#)
`altlongragged4colheader` (style) [295](#)
`altlongragged4colheaderborder`
(style) [296](#)
`\altnewglossary` [73](#), [412](#), [662](#), [918](#)
`altsuper4col` (style) [308](#)
`altsuper4colborder` (style) [309](#)
`altsuper4colheader` (style) [308](#)
`altsuper4colheaderborder` (style) [309](#)
`altsuperragged4col` (style) [314](#)
`altsuperragged4colborder` (style) [315](#)
`altsuperragged4colheader` (style) [315](#)
`altsuperragged4colheaderborder`
(style) [315](#)
`almtree` (style) [322](#)
`almtreegroup` (style) [325](#)
`almtreehypergroup` (style) [325](#)
`\andname` [45](#),
[211](#), [396](#), [491](#), [646](#), [743](#), [902](#), [999](#)
`\AnyTrackedLanguages` [46](#), [326](#), [327](#),
[397](#), [572](#), [589](#), [647](#), [825](#), [842](#), [903](#)
`\appto` [22](#),
[27](#), [79](#), [88](#), [89](#), [204](#), [328](#), [338](#),

<code>\csedef</code>	110,	190–192, 218, 220, 222, 223,
	203, 433, 488, 684, 740, 941, 995	225, 227, 248, 328, 397, 399,
<code>\csgdef</code>	19, 51, 71, 74, 109,	411, 416, 417, 421, 422, 441,
	110, 211, 226, 399, 411, 412,	442, 480, 481, 495, 497–499,
	432, 433, 492, 501, 649, 661,	501, 502, 507, 513, 579–588,
	662, 683, 684, 744, 753, 885,	590, 647, 649, 661, 666, 667,
	905, 917, 918, 940, 941, 999, 1008	672, 692, 732, 733, 747, 749–
<code>\cslet</code>	78, 79, 94, 101, 221,	751, 753, 754, 759, 765, 833–
	415, 424, 429, 498, 665, 675,	837, 839–843, 903, 904, 917,
	680, 750, 920, 921, 931, 936, 1005	920, 922, 923, 928, 971, 987,
<code>\csname</code>	15–18, 26, 41, 44,	988, 1003, 1004, 1006–1009, 1019
	46, 47, 52, 53, 56, 58, 59, 62,	<code>\csxdef</code>
	64, 67, 72–74, 81, 82, 88–90,	99, 110,
	92, 93, 97, 98, 100–104, 106,	427, 433, 678, 684, 774, 934, 941
	107, 123, 125, 126, 136–141,	<code>\currentglossary</code>
	157–164, 175, 176, 180–182,
	187–190, 196, 199, 200, 202,	14, 15, 50, 214, 270,
	203, 208–210, 213–215, 218,	271, 273, 379, 399, 493, 628,
	235, 236, 240, 263–269, 322,	629, 649, 745, 883, 904, 1001, 1031
	324, 328, 339, 340, 343, 347,	<code>\currentglssubentry</code>
	353–355, 372, 373, 379–381,	15,
	385, 394, 396, 397, 400, 402,	229, 379, 503, 629, 755, 883, 1010
	404, 406–408, 411, 412, 416,	<code>\CurrentOption</code>
	420–423, 426–431, 441, 443–	41, 327, 334,
	445, 452–455, 461–465, 470,	394, 537, 589, 644, 790, 842, 900
	472–474, 477–479, 484, 486–	<code>\CurrentTrackedLanguage</code>
	488, 490, 491, 493–495, 501,
	504, 505, 507, 509, 535–538,	46, 326, 327,
	540–543, 546, 553, 554, 571–	397, 572, 589, 647, 826, 842, 903
	574, 585, 590, 594, 595, 622,	<code>\CurrentTrackedTag</code> .
	623, 629–631, 635, 644, 646,	46, 326, 327,
	647, 650, 652, 654, 656, 657,	397, 572, 589, 647, 825, 842, 903
	659, 661, 662, 666, 667, 670–	<code>\CustomAcronymFields</code>
	674, 677–682, 692, 694–696,
	704–706, 712–717, 721, 722,	263, 534, 787, 788
	724, 725, 729–731, 735, 738–	<code>\CustomNewAcronymDef</code>
	740, 742–747, 753, 756–759,
	761, 774, 788–791, 794–796,	263, 534, 535, 787, 788
	799, 806–808, 824–827, 838,	D
	843, 847, 848, 875, 876, 883–	<code>\d</code>
	885, 889, 900–903, 906, 908,	26, 385, 635, 889
	910, 912–914, 917, 918, 922,	<code>\day</code>
	923, 926–930, 933–937, 939,	180, 184, 472, 475, 574,
	948, 949, 958–960, 966–971,	577, 724, 727, 828, 830, 978, 981
	976, 977, 979, 980, 984–986,	<code>debug</code> (option)
	991, 993–995, 997, 998, 1000,	7
	1001, 1003, 1013, 1015, 1029–1031	<code>\DeclareAcronymList</code>
<code>\csshow</code>	268, 537, 790, 1030	20,
<code>\csuse</code>	47,	21, 22, 245, 247, 381–383, 511,
	50, 71, 78, 81, 82, 89, 90, 165,	512, 522, 524, 525, 527, 529,
		531, 533, 535, 631–633, 763,
		764, 774, 776, 778, 780, 782,
		784, 786, 788, 886, 887, 1018, 1019
		<code>\DeclareCurrentRelease</code>
	
		4, 269, 273, 276, 280,
		286, 291, 297, 302, 310, 316,
		326, 327, 334, 373, 374, 624, 877
		<code>\DeclareListParser</code> 194, 483, 734, 989
		<code>\DeclareOption</code>
		11, 327, 334,
		377, 537, 589, 627, 790, 842, 881
		<code>\DeclareOptionX</code> . . .
		11, 377, 627, 881

<code>\DeclareRelease</code>	<code>\defglentryfmt</code>
..... 4, 269, 273, 276, 280,	73, 74, 247, 260, 411, 412, 441,
286, 291, 297, 302, 310, 316,	512, 521, 522, 524, 526, 528,
326, 327, 334, 373, 374, 624, 877	530, 532, 534, 661–663, 692,
<code>\DeclareRobustCommand</code>	764, 773, 775, 777, 778, 780,
..... 47, 397, 530, 648, 782, 903	783, 785, 787, 917, 918, 1019, 1028
<code>\def</code>	<code>\define@boolkey</code>
11, 12, 14–	10, 12–
21, 25–27, 32, 33, 35, 37–40,	15, 20, 27, 28, 31–33, 38, 120,
42–44, 47, 48, 51, 55, 57–63,	228, 377–379, 381, 386, 387,
67, 71–80, 82, 92, 94–99, 101,	389, 390, 392, 442, 502, 626–
104, 105, 109–113, 119, 120,	629, 631, 636, 639, 640, 642,
123–134, 136–164, 175, 176,	693, 754, 881–883, 886, 890,
179, 184, 186–188, 190–193,	891, 893, 894, 897, 898, 946, 1009
195–198, 200, 203, 204, 207–	<code>\define@choicekey</code>
210, 213, 214, 216, 221–226,	7,
228, 234–238, 240, 243–245,	9, 10, 15, 28, 30, 32, 36, 39,
273–275, 301, 302, 322–325,	41, 78, 226, 227, 375–377, 379,
327, 328, 330–334, 336–339,	387, 388, 390, 391, 393, 414,
353–360, 377–382, 385, 386,	501, 502, 624, 626, 627, 629,
390, 392–399, 401, 403–409,	637–639, 642, 643, 664, 753,
411–415, 417, 423–430, 433–	754, 878, 880, 881, 883, 891,
435, 441–465, 470, 472, 475–	892, 894, 896, 898, 899, 920, 1009
482, 484–486, 488–491, 493,	<code>\define@key</code>
494, 498–502, 506–511, 522–	11, 14, 22, 23, 29,
527, 529–531, 533, 535, 538–	31, 33, 75–80, 88, 89, 119, 120,
540, 553–558, 567–571, 574,	176, 226, 228, 327, 336–338,
577, 579, 584, 585, 589, 591–	377, 379, 383, 387, 390, 413–
596, 610, 611, 622, 623, 627–	415, 420, 442, 470, 501, 502,
632, 634–636, 640–643, 645–	538, 539, 589, 627, 629, 633,
649, 651, 653–659, 661–665,	637, 640, 663–665, 671, 693,
667, 673, 675–681, 683–686,	722, 753, 754, 792, 793, 842,
692–717, 721, 722, 724, 726,	881, 883, 887, 891, 893, 894,
728–733, 736–738, 740–746,	919–921, 927, 946, 977, 1008–1010
750–754, 758–763, 774, 776–	<code>\DefineAcronymSynonyms</code>
784, 786, 788, 791–794, 806–
811, 821–824, 827, 828, 830,	42, 258, 259,
832, 838, 842–846, 848, 849,	395, 521, 645, 773, 900, 1027, 1028
863, 864, 875–877, 881–887,	<code>\delimN</code>
889, 890, 894, 896–901, 903–	50, 183, 194, 225,
905, 907, 909–914, 917–921,	237, 238, 399, 474, 483, 500,
923, 930–938, 940–943, 946,	508, 576, 649, 726, 734, 752,
948–971, 976–978, 981, 983–	760, 830, 904, 980, 990, 1008, 1014
993, 995–1000, 1002, 1005,	<code>\delimR</code>
1006, 1008–1010, 1012–1018, 1031	50,
<code>\def@glsex@dycheckbackslash</code>	183, 237, 399, 474, 508, 576,
133, 134, 450, 451, 702, 703, 956	649, 726, 760, 830, 904, 980, 1014
<code>\DefaultNewAcronymDef</code>	<code>description (key)</code>
260,	75
371, 521, 522, 567, 773, 775, 820	<code>description (option)</code>
<code>\defgldisplay</code>	32
119, 441, 692	<code>descriptionaccess (key)</code>
<code>\defgldisplayfirst</code> ...	337
119, 441, 692	<code>\DescriptionDUANewAcronymDef</code> ..

	261, 524, 526, 777, 778
	<code>\DescriptionFootnoteNewAcronymDef</code>

	260,
	371, 523, 524, 568, 775, 776, 821

<code>\descriptionname</code> . . .	44 , 47 , 281 – 285 , 287 , 288 , 293 , 294 , 296 , 304 , 306 – 309 , 311 – 316 , 396 , 397 , 599 – 602 , 605 – 607 , 612 – 614 , 616 – 618 , 646 , 647 , 852 – 856 , 858 – 861 , 865 – 872 , 901 , 903	<code>\doifglossarynoexistsordo</code> . .	66 , 72 , 408 , 411 , 658 , 661 , 914 , 917
<code>\DescriptionNewAcronymDef</code> .	261 , 371 , 526 , 527 , 568 , 779 , 780 , 822	<code>\dtl@ifsingle</code>	506 , 758
<code>descriptionplural</code> (key)	75	<code>\dtl@insertinto</code> .	218 , 496 , 748 , 1003
<code>descriptionpluralaccess</code> (key) . .	337	<code>\dtl@sortresult</code> 218 , 219 , 496 , 748 , 1003 , 1004
<code>\DH</code>	27 , 386 , 636 , 890	<code>\dtlcompare</code>	219 , 496 , 748 , 1004
<code>\dh</code>	27 , 386 , 636 , 890	<code>\dtlicompare</code>	219 , 496 , 748 , 1004
<code>\dimen@</code>	249 , 290 , 322 , 514 , 603 , 621 , 622 , 766 , 857 , 875 , 1021	<code>\DTLifinlist</code>	75 , 123 , 413 , 443 , 663 , 694 , 919 , 948
<code>\directlua</code>	42 , 900	<code>\DTLifint</code>	234 , 506 , 758 , 1013
<code>\disable@keys</code>	42 , 394 , 644 , 900	<code>\dtlletterindexcompare</code> 219 , 496 , 748 , 1003
<code>disablemakegloss</code> (option)	34	<code>\DTLsubstituteall</code>	127 , 446 , 696 , 950
<code>\do</code>	31 , 41 , 43 , 45 , 55 , 56 , 63 , 64 , 86 , 87 , 127 , 175 , 180 – 182 , 192 , 194 , 195 , 202 , 210 , 217 , 247 , 389 , 394 , 395 , 401 , 402 , 407 , 418 , 419 , 445 , 470 , 472 , 473 , 482 , 483 , 488 , 491 , 495 , 512 , 522 , 524 , 526 , 527 , 529 , 532 , 533 , 535 , 574 , 594 , 595 , 639 , 644 , 646 , 652 , 657 , 669 , 670 , 696 , 721 , 724 , 725 , 734 , 735 , 740 , 743 , 746 , 747 , 764 , 774 , 777 , 778 , 780 , 782 , 785 , 786 , 788 , 828 , 847 , 848 , 893 , 900 – 902 , 907 , 908 , 913 , 925 , 926 , 950 , 976 , 978 – 980 , 988 – 990 , 995 , 999 , 1002 , 1019 , 1031	<code>\dtlwordindexcompare</code> 218 , 496 , 748 , 1003
<code>\do@glo@storeentry</code> 16 – 19 , 100 , 379 – 381 , 428 , 629 – 631 , 679 , 883 – 885 , 935	<code>dua</code> (acrstyle)	253 , 365
<code>\do@gls@link@checkfirsthyper</code> 122 , 124 , 136 – 141 , 157 – 164 , 353 – 355 , 443 , 452 – 455 , 461 – 465 , 553 , 554 , 693 , 694 , 704 – 707 , 712 – 716 , 806 , 807 , 947 , 948 , 957 – 960 , 966 – 971	<code>dua</code> (option)	32
<code>\do@gls@xdycheckbackslash</code> 126 , 127 , 445 , 696 , 950	<code>dua-desc</code> (acrstyle)	255 , 367
<code>\do@gls@disablehyperinlist</code> 123 , 124 , 443 , 694 , 948	<code>\DUANewAcronymDef</code> 262 , 532 , 533 , 785 , 786
<code>\do@gls@haschildren</code> 67 , 408 , 409 , 658 , 659 , 914	E	
<code>\do@write@gls@labelname</code> 35 , 35 , 36 , 641 , 896	<code>\eappto</code>	412 , 413 , 429 , 488
		<code>\edef</code>	43 , 46 , 54 – 56 , 58 , 60 , 61 , 72 , 81 , 82 , 85 , 87 , 125 – 134 , 178 , 184 – 188 , 190 – 192 , 194 , 200 , 216 , 218 , 219 , 223 , 234 , 238 , 240 , 381 , 382 , 395 , 397 , 401 – 408 , 411 – 413 , 416 – 419 , 422 – 425 , 429 , 441 , 443 – 451 , 469 – 471 , 475 – 478 , 480 , 481 , 483 , 486 , 489 , 490 , 492 , 494 – 496 , 499 , 503 , 506 , 508 , 509 , 521 , 523 , 524 , 526 , 528 , 530 , 532 , 567 – 570 , 573 – 575 , 577 , 594 , 646 , 647 , 651 – 656 , 661 , 662 , 666 , 667 , 669 , 670 , 692 , 695 – 703 , 722 , 723 , 726 , 728 – 730 , 732 – 734 , 738 , 741 , 742 , 746 , 748 , 751 , 758 , 760 , 761 , 773 , 775 , 777 , 779 , 781 , 783 , 785 , 820 – 823 , 826 – 828 , 830 , 901 , 902 , 907 – 912 , 917 , 922 , 923 , 925 , 926 , 948 – 956 , 977 , 981 , 983 – 989 , 993 , 1001 , 1003 , 1004 , 1007 , 1013 , 1015
		<code>\egroup</code> .	26 , 94 , 175 , 208 , 215 , 217 , 238 , 385 , 424 , 470 , 493 , 495 ,

635, 675, 722, 745, 747, 889, 931, 976, 998, 1001, 1002, 1015	<code>\end@glislistofacronyms</code>
<code>\else</code> 8, 13–15, 18–21, 23, 25, 27, 28, 35, 37, 38, 41–43, 48, 49, 51– 58, 60–63, 71, 78, 81, 97, 98, 101–104, 109, 110, 120, 123– 134, 167, 178–180, 183–186, 188–190, 192, 193, 200, 202– 204, 207–210, 214, 223, 228– 230, 235, 237, 238, 249, 262, 263, 277, 281, 282, 284, 287, 288, 290, 292, 294, 295, 304, 305, 307, 311, 313, 315, 318, 320, 321, 323, 324, 344, 375, 376, 378, 379, 381–384, 386, 387, 391–395, 398–407, 414, 416, 426, 427, 429, 430, 433, 443–455, 466, 471, 472, 474– 481, 486–491, 493, 499, 502– 504, 506, 508, 514, 524, 525, 527, 529, 530, 532–535, 543, 574–579, 583–585, 594, 597, 599, 600, 602–606, 611–613, 615–617, 619–623, 625, 628– 634, 636, 637, 641–646, 648– 657, 664, 666, 677, 678, 680, 681, 684, 694–706, 717, 718, 723, 724, 726, 728–733, 738– 743, 745, 751, 754–756, 758, 760, 766, 776–778, 780, 782– 785, 787, 788, 797, 827, 828, 830–832, 836–838, 847, 850, 852–860, 865–867, 869–871, 873–877, 879, 882–888, 890, 891, 896, 897, 900, 901, 904– 913, 917, 920, 922, 933, 934, 936, 937, 941, 946–956, 972, 978, 979, 981–983, 985–989, 994–999, 1001, 1006, 1010, 1011, 1013–1015, 1021, 1028, 1031 21, 382, 383, 632, 886, 887
<code>\emph</code> 210, 239, 491, 509, 743, 761, 999, 1015	<code>\EndAccSupp</code> 335, 538, 791
<code>\empty</code> 238, 335, 508, 537, 760, 791, 1015	<code>\endcsname</code> 15–18, 26, 41, 44, 46, 47, 52, 53, 56, 58, 59, 62, 64, 67, 72–74, 81, 82, 88–90, 92, 93, 97, 98, 100–104, 106, 107, 123, 125, 126, 136–141, 157–164, 175, 176, 180–182, 187–190, 196, 199, 200, 202, 203, 208–210, 213–215, 218, 235, 236, 240, 263–269, 322, 324, 328, 339, 340, 343, 347, 353–355, 372, 373, 379–381, 385, 394, 396, 397, 400, 402, 404, 406–408, 411, 412, 416, 420–423, 426–431, 441, 443– 445, 452–455, 461–465, 470, 472–474, 477–479, 484, 486– 488, 490, 491, 493–495, 501, 504, 505, 507, 509, 535–538, 540–543, 546, 553, 554, 571– 574, 585, 590, 594, 595, 622, 623, 629–631, 635, 644, 646, 647, 650, 652, 654, 656, 657, 659, 661, 662, 666, 667, 670– 674, 677–682, 692, 694–696, 704–706, 712–717, 721, 722, 724, 725, 729–731, 735, 738– 740, 742–747, 753, 756–759, 761, 774, 788–791, 794–796, 799, 806–808, 824–827, 838, 843, 847, 848, 875, 876, 883– 885, 889, 900–903, 906, 908, 910, 912–914, 917, 918, 922, 923, 926–930, 933–937, 939, 948, 949, 958–960, 966–971, 976, 977, 979, 980, 984–986, 991, 993–995, 997, 998, 1000, 1001, 1003, 1013, 1015, 1029–1031
<code>\encodingdefault</code> . . 26, 385, 635, 889	<code>\endfoot</code> 281–283, 285, 287, 288, 292–294, 296, 599– 602, 605–607, 852–856, 858–861
<code>\end</code> 182, 222, 277, 280–283, 285, 286, 288–316, 474, 499, 576, 597–601, 603–618, 725, 750, 829, 850, 852–872, 980, 1006	<code>\endgroup</code> 6, 8, 35, 202, 207, 238, 374, 376, 487, 489, 508, 624, 626, 641, 739, 741, 760, 878, 880, 896, 994, 997, 1015
<code>\end@doifinlist</code> . . . 55, 401, 651, 907	<code>\endhead</code> 281–285, 287, 288, 292–294, 296, 599– 602, 605–607, 852–856, 858–861
<code>\end@getprefix</code> . . . 209, 490, 742, 998	

<code>\endtheglossary</code>	6, 374, 624, 878	
<code>entrycounter</code> (option)	<u>14</u>	997, 999, 1001, 1003, 1006, 1007, 1011, 1013, 1015, 1029–1032
<code>\entryname</code>	<u>44</u> , 47, 281– 285, 287, 288, 293, 294, 296, 304, 306–309, 311–316, 396, 397, 599–602, 605–607, 612– 614, 616–618, 646, 647, 852– 856, 858–861, 865–872, 901, 903	<code>\expandonce</code> 81, 82, 93, 127, 128, 187, 188, 203, 218, 219, 246, 416, 417, 445, 446, 477–479, 488, 496, 504, 505, 511, 521, 523, 525, 527, 528, 531, 532, 538, 666, 667, 674, 696, 697, 729, 730, 740, 748, 756, 757, 763, 773–777, 779, 781, 784, 785, 791, 922, 923, 930, 950, 951, 984, 985, 995, 1003, 1004, 1018
environments:		
<code>theglossary</code>	<u>230</u>	
<code>\epreto</code>	<u>104</u> , <u>937</u>	
<code>\equal</code>	29, 43, 53, 125, 194, 387, 395, 400, 444, 482, 506, 594, 637, 645, 650, 695, 734, 758, 847, 891, 901, 906, 948, 989, 1031	<code>\ExplSyntaxOff</code> 6, 234, 235, 273, 878, 1012, 1013
esclocations (option)	<u>12</u>	<code>\ExplSyntaxOn</code> 4, 234, 270, 877, 1012, 1013
<code>\exp</code>	<u>273</u>	
<code>\expandafter</code>	8, 16–18, 25, 26, 41, 43, 46, 56, 58, 59, 61–64, 67, 72–75, 81, 82, 86– 90, 92, 93, 97, 98, 100, 102– 107, 118, 119, 123, 125–133, 165–167, 170, 175, 178, 182, 186–189, 200, 202–204, 207, 208, 211, 214, 218, 223, 224, 231, 236, 238, 240, 263–269, 322, 328, 339, 340, 343, 344, 371–373, 376, 380, 381, 385, 394–397, 402, 404–408, 411– 413, 416–423, 426–431, 441, 443–451, 466, 470, 471, 473, 474, 476–479, 486–491, 493, 495, 499, 500, 504, 505, 507– 509, 523, 528, 535–538, 540– 544, 567, 571–574, 576, 577, 590, 594, 595, 622, 625, 629– 631, 634, 635, 644, 646, 647, 652, 654, 656, 657, 659, 661– 663, 666, 667, 669–674, 677– 682, 692, 694–702, 717–719, 722, 723, 725, 728–731, 738– 743, 745, 747, 751, 752, 756, 757, 759–761, 774, 775, 781, 788–791, 794–797, 820, 824– 827, 829, 830, 843, 847, 848, 875, 880, 884, 885, 889, 900– 902, 908, 910–914, 917–919, 922, 923, 925–930, 933–939, 946, 948–956, 971, 972, 974, 976, 977, 980, 983–986, 993–	F <code>\fi</code> 7, 8, 10, 11, 13–16, 18–21, 23, 25, 27, 28, 30–32, 34, 35, 37, 38, 40–43, 47–49, 52–63, 71, 74, 78, 81, 93, 97–99, 101–104, 109, 110, 120, 123–134, 167, 177–180, 183, 185, 186, 188, 189, 191– 195, 200–205, 207–211, 214, 216, 223, 227–230, 235, 237, 238, 249, 259, 262, 263, 277, 281, 282, 284, 287, 288, 290– 292, 294, 295, 304, 305, 307, 311, 313, 315, 318–324, 335, 344, 374–387, 389, 391–395, 397–407, 412, 414, 416, 426– 430, 433, 443–455, 466, 470– 472, 474, 476–480, 482, 483, 487–494, 499, 502–504, 506– 509, 514, 521, 522, 524, 525, 527–535, 537, 538, 544, 573– 579, 583–585, 594, 597, 599, 600, 602–606, 611–613, 615– 617, 619–634, 636–638, 641– 646, 648–657, 662, 664, 666, 674, 676–681, 684, 694–707, 718, 722–724, 726, 728–735, 738–746, 751, 754–756, 758– 760, 766, 773, 774, 776– 778, 780, 782–788, 790, 791, 797, 826–828, 830–832, 836– 838, 847, 850, 852–860, 865– 867, 869–871, 873–888, 890,

891, 893–901, 903–913, 917, 918, 920, 922, 930, 932–937, 940, 941, 946–956, 972, 977– 979, 981–983, 985–990, 994– 1002, 1007, 1009–1011, 1013– 1015, 1021, 1028, 1029, 1031	<code>\forlistloop</code> 198, 224, 485, 500, 737, 752, 992, 1008
<code>\findrootlanguage</code> 62, 406, 656	
<code>first</code> (key) 76	
<code>firstaccess</code> (key) 336	
<code>\firstacronymfont</code> 118, 245, 248–250, 255, 256, 352, 353, 362, 363, 368, 441, 511, 513, 514, 519, 523, 526, 528, 530, 553, 559, 560, 564, 691, 692, 763, 765, 766, 771, 775, 779, 781, 783, 806, 812, 813, 817, 946, 1017, 1020, 1021, 1025, 1026	G
<code>firstplural</code> (key) 76	<code>\gdef</code> 17, 35, 56, 72, 92, 97, 98, 201, 228, 272, 380, 402, 411, 422, 426, 487, 502, 503, 594, 630, 641, 652, 661, 673, 677, 739, 754, 847, 884, 896, 908, 917, 929, 933, 994, 1010, 1031
<code>firstpluralaccess</code> (key) 336	<code>\Genacrfullformat</code> 118, 118, 246, 248–250, 256, 352, 353, 362, 363, 368, 440, 441, 512–514, 519, 552, 553, 559, 560, 564, 691, 763–766, 771, 805, 806, 812, 813, 817, 945, 946, 1018, 1020, 1021, 1025
<code>\fmtversion</code> 27, 386, 636, 890	<code>\genacrfullformat</code> 117, 118, 118, 246, 248–250, 255, 352, 352, 362, 368, 440, 441, 511–514, 519, 552, 559, 560, 564, 691, 692, 763–766, 771, 805, 806, 812, 813, 817, 945, 946, 1018, 1020, 1021, 1025
<code>\footnote</code> 255, 256, 260, 368, 519, 522, 564, 771, 775, 817, 1025, 1026, 1028	<code>\GenericAcronymFields</code> . 246, 246, 248–255, 257, 258, 362–368, 370, 371, 511–521, 559–561, 563, 564, 566, 763–772, 812– 817, 819, 820, 1018–1025, 1027
<code>footnote</code> (acrstyle) 255, 368	<code>\Genplacrfullformat</code> 117, 119, 246, 248–250, 256, 352, 353, 362, 363, 368, 440, 441, 512–514, 519, 552, 553, 559, 560, 564, 691, 692, 764– 766, 771, 805, 806, 812, 813, 817, 945, 946, 1018–1021, 1026
<code>footnote</code> (option) 31	<code>\genplacrfullformat</code> 117, 118, 119, 246, 248–250, 256, 352, 353, 362, 363, 368, 440, 441, 512–514, 519, 552, 553, 559, 560, 564, 691, 692, 764– 766, 771, 805, 806, 812, 813, 817, 945, 946, 1018–1021, 1026
<code>footnote-desc</code> (acrstyle) 257, 370	<code>\GetTitleStringDisableCommands</code> 276, 850
<code>footnote-sc</code> (acrstyle) 256, 369	<code>\GetTitleStringSetup</code> 276, 850
<code>footnote-sc-desc</code> (acrstyle) . . 257, 370	<code>\glo@desc</code> 579, 832
<code>footnote-sm</code> (acrstyle) 257, 370	<code>\glo@do@compare</code> 218, 219, 496, 748, 1003, 1004
<code>footnote-sm-desc</code> (acrstyle) . . 258, 370	
<code>\FootnoteNewAcronymDef</code> 261, 371, 528, 529, 569, 781, 782, 823	
<code>\footnotesize</code> 8, 376, 625, 879	
<code>\forallacronyms</code> . . . 63, 407, 657, 913	
<code>\forallglossaries</code> . . 63, 64, 200, 213, 322, 407, 486, 492, 621, 657, 738, 744, 875, 913, 993, 1000	
<code>\forallglsentries</code> 35, 64, 107, 110, 177, 391, 407, 431, 432, 434, 471, 641, 657, 682– 684, 722, 896, 913, 939–941, 977	
<code>\ForEachTrackedDialect</code> 46, 326, 327, 397, 572, 589, 647, 825, 842, 903	
<code>\forglsentries</code> 63, 64, 67, 101, 221, 322, 407, 408, 429, 498, 621, 657, 659, 680, 750, 875, 913, 914, 936, 1005	
<code>\forallistcsloop</code> 217, 222, 495, 499, 746, 750, 1002, 1006	

<code>\glo@grabfirst</code>	222 , 224 , 499 , 500 , 751 , 752 , 1006 , 1007	738 , 746 , 847 , 878 , 879 , 887 , 888 , 990 , 991 , 993 , 994 , 1002 , 1031	
<code>\glo@label</code>	67 , 101 , 408 , 429 , 659 , 680 , 914 , 936	<code>\glossary</code>	38 , 39 , 39 , 392 , 393 , 574 , 642 , 643 , 827 , 898
<code>\glo@list</code>	101 , 429 , 680 , 936	<code>\glossaryentry</code> ...	208 , 210 , 490 , 491 , 574 , 742 , 743 , 827 , 997 , 998
<code>\glo@name</code>	231 , 504 , 756 , 1011	<code>glossaryentry (counter)</code>	14
<code>\glo@parent</code>	67 , 408 , 659 , 914	<code>\glossaryentryfield</code> 232 , 361 , 504 ,	505 , 559 , 579 – 584 , 586 – 588 , 756 , 757 , 812 , 832 – 837 , 839 – 841
<code>\glo@type</code>	101 , 429 , 680 , 936	<code>\glossaryentrynumbers</code>	11 , 12 , 182 , 214 , 215 , 224 , 228 , 377 , 378 , 474 , 493 , 500 , 502 , 503 , 576 , 627 , 726 , 745 , 751 , 752 , 754 , 755 , 829 , 881 , 980 , 1000 , 1001 , 1007 , 1010
<code>\glo@value</code>	86 , 418 , 669 , 925	<code>\glossaryheader</code> ..	182 , 222 , 231 , 274 , 277 – 279 , 281 – 285 , 287 , 288 , 292 – 296 , 298 – 301 , 303 , 305 , 306 , 310 , 312 , 314 , 317 , 319 – 321 , 323 , 325 , 474 , 499 , 504 , 576 , 595 , 597 – 602 , 604 – 613 , 615 – 617 , 619 – 623 , 725 , 750 , 756 , 829 , 848 , 850 – 856 , 858 – 867 , 869 – 877 , 980 , 1006 , 1011
<code>\global</code>	17 , 18 , 81 , 84 , 94 , 100 , 106 , 107 , 202 , 215 , 223 , 224 , 228 , 290 , 291 , 380 , 381 , 416 , 418 , 424 , 428 , 431 , 487 , 493 , 499 , 500 , 503 , 604 , 630 , 631 , 666 , 668 , 675 , 679 , 682 , 739 , 745 , 751 , 752 , 755 , 857 , 884 , 885 , 922 , 924 , 931 , 935 , 939 , 994 , 1001 , 1007 , 1010	<code>\glossarymark</code> 51 , 52 , 399 , 649 , 650 , 905	
<code>\glo@linkprefix</code>	124 , 124 , 176 , 231 , 444 , 470 , 504 , 695 , 722 , 756 , 948 , 977 , 1011	<code>\glossaryname</code>	19 , 44 , 46 , 47 , 381 , 396 , 397 , 631 , 646 , 647 , 885 , 901 – 903
<code>\glosortentrieswarning</code>	23 , 23 , 216 , 383 , 384 , 494 , 633 , 746 , 887 , 888 , 1002	<code>\glossarypostamble</code>	51 , 182 , 222 , 399 , 474 , 499 , 576 , 649 , 725 , 751 , 829 , 905 , 980 , 1006
<code>\glossaries</code>	270 – 272	<code>\glossarypreamble</code>	50 , 182 , 222 , 399 , 473 , 498 , 576 , 649 , 725 , 750 , 829 , 904 , 980 , 1006
<code>\GlossariesWarning</code> 7 , 9 , 23 , 23 , 24 ,	27 – 29 , 34 – 36 , 39 , 45 , 51 , 54 , 66 , 70 , 71 , 78 , 81 , 84 , 85 , 111 – 113 , 174 , 189 – 192 , 196 – 199 , 202 , 204 , 209 , 212 , 213 , 262 , 335 , 375 , 377 , 383 , 384 , 386 , 387 , 390 – 393 , 399 , 401 , 408 , 410 , 414 , 416 , 434 , 435 , 441 , 442 , 469 , 480 , 481 , 483 – 487 , 490 , 492 , 493 , 505 , 507 , 537 , 573 , 624 – 626 , 633 , 634 , 636 , 637 , 640 – 643 , 649 , 651 , 658 , 661 , 664 , 666 , 668 , 669 , 685 , 692 , 693 , 721 , 732 , 733 , 735 – 739 , 742 , 744 , 745 , 757 , 759 , 786 , 791 , 826 , 878 – 880 , 887 , 888 , 890 , 891 , 895 , 896 , 898 , 902 , 905 , 907 , 914 , 916 , 917 , 920 , 922 , 924 , 925 , 942 , 976 , 986 – 988 , 990 – 995 , 998 , 1000 , 1028	<code>\glossarysection</code>	51 , 182 , 222 , 399 , 473 , 498 , 576 , 649 , 725 , 750 , 829 , 905 , 980 , 1006
<code>\GlossariesWarningNoLine</code>	7 , 23 , 23 , 195 , 196 , 200 , 216 , 272 , 375 , 383 , 384 , 482 – 484 , 486 , 494 , 594 , 624 , 625 , 633 , 735 , 736 ,	<code>\glossarystyle</code>	235 , 507 , 759
		<code>glossarysubentry (counter)</code>	15
		<code>\glossarysubentryfield</code>	233 , 361 , 505 , 506 , 559 , 579 – 588 , 757 , 812 , 832 – 841
		<code>\glossarytitle</code> ...	182 , 213 , 214 , 222 , 226 , 473 , 493 , 498 , 501 , 576 , 725 , 744 , 745 , 750 , 753 , 829 , 980 , 1000 , 1001 , 1006 , 1008
		<code>\glossarytoctitle</code>	11 , 19 , 20 , 40 , 44 , 47 , 51 , 182 , 213 , 222 , 226 , 227 , 377 , 381 ,

382, 393, 394, 396, 397, 399,
 473, 493, 498, 501, 502, 576,
 627, 631, 632, 644, 646, 647,
 649, 725, 744, 750, 753, 754,
 829, 881, 885, 886, 899, 901,
 903, 905, 980, 1000, 1006, 1009
 \glossentry
 101, 215, 224, 231, 274, 277–
 282, 284, 292, 293, 295, 303,
 305, 307, 310, 312, 314, 318,
 319, 321, 323, 429, 493, 500,
 505, 595, 597–600, 604–606,
 611–613, 615–617, 619–622,
 680, 745, 752, 757, 848, 850–
 854, 858–860, 865–867, 869–
 874, 876, 936, 1001, 1007, 1011
 \Glossentrydesc
 232, 361, 505, 558, 757, 811, 1012
 \glossentrydesc
 . 232, 232, 274, 275, 277–279,
 281, 282, 284, 292, 293, 295,
 303, 305, 307, 310–312, 314,
 318, 319, 321, 323, 324, 361,
 505, 558, 596–600, 604–606,
 611–613, 615–617, 619–623,
 756, 811, 849–854, 858–860,
 865–867, 869–871, 873–876, 1012
 \Glossentryname . 231, 504, 756, 1012
 \glossentryname 231,
 231, 233, 274–282, 284, 292,
 293, 295, 303, 305, 307, 310,
 312, 314, 318, 319, 321, 323,
 324, 360, 361, 504, 558, 595–
 600, 604–606, 611–613, 615–
 617, 619–623, 756, 811, 849–
 854, 858–860, 865–867, 869–
 871, 873, 874, 876, 1011, 1012
 \Glossentrysymbol
 232, 361, 505, 558, 757, 812, 1012
 \glossentrysymbol
 232, 232, 274, 275,
 284, 295, 307, 314, 318, 319,
 321, 323, 324, 361, 505, 558,
 595, 596, 600, 606, 613, 617,
 619–623, 757, 812, 849, 854,
 860, 867, 871, 873, 874, 876, 1012
 \GLS 137, 453, 704, 958
 \Gls 111, 136, 136, 240,
 259, 434, 452, 509, 521, 685,
 704, 761, 773, 942, 958, 1015, 1028
 \gls ... 39, 111, 136, 136, 196, 230,
 240, 259, 393, 434, 452, 483,
 503, 509, 521, 643, 685, 704,
 735, 755, 761, 773, 898, 942,
 957, 958, 990, 1011, 1015, 1028
 \gls@delimN 237, 238, 1014
 \gls@accessibility
 335, 336, 538, 791, 792
 \gls@accsupp@engine 335, 335, 538, 791
 \gls@Alphpage 202, 203,
 207, 488, 489, 739–741, 995, 997
 \gls@alphpage 202, 203,
 207, 488, 489, 739–741, 995, 997
 \gls@arabicpage 202, 203,
 207, 488, 489, 739–741, 995, 996
 \gls@assign@desc 94,
 94, 99, 424, 428, 675, 679, 931, 935
 \gls@assign@desc@field 24
 \gls@assign@descplural 522, 529,
 531, 533, 567, 570, 571, 774,
 781, 782, 784, 786, 821, 823, 824
 \gls@assign@descplural@field .. 25
 \gls@assign@field
 83, 83, 88, 89, 94,
 96, 98–100, 328, 417, 420, 424,
 425, 427, 428, 590, 667, 671,
 675–679, 843, 923, 927, 931–935
 \gls@assign@firstpl 522–525, 527,
 529, 531–533, 567–571, 773,
 774, 776–782, 784, 786, 820–824
 \gls@assign@name@field 25
 \gls@assign@plural 522–525,
 527, 529, 531–533, 567–571,
 774, 776–782, 784, 786, 821–824
 \gls@assign@symbolplural
 522–525, 527,
 531–533, 568, 569, 571, 774,
 776–780, 784, 786, 821, 822, 824
 \gls@assign@type@field 24
 \gls@automake@makegloss
 36, 36, 37, 38, 193, 896, 897, 989
 \gls@automake@nr 36, 36, 37, 38, 192,
 391, 482, 642, 733, 896, 897, 988
 \gls@automake@option
 36, 36, 37, 896, 897
 \gls@automake@val
 36, 37, 391, 642, 896, 897
 \gls@begindocdefs 85, 85, 418, 669, 925
 \gls@checkisacronymlist ... 22,
 122, 383, 443, 632, 694, 887, 947

<code>\gls@checkseeallowed</code>	77, 77, 194, 196, 414, 417, 482, 483, 664, 734, 735, 920, 989, 991	<code>\gls@ifnotmeasuring</code>	104, 105, 106, 202, 430, 431, 681, 682, 937–939, 994
<code>\gls@checkseeallowed@preambleonly</code>	78, 414, 417, 664, 920	<code>\gls@inlinepostchild</code>	274, 275, 579, 595, 596, 832, 848, 849
<code>\gls@codepage</code>	33, 62, 190, 191, 216, 390, 406, 480, 481, 494, 640, 656, 657, 732, 733, 746, 894, 912, 986, 987, 1002	<code>\gls@inlinesep</code>	274, 579, 595, 596, 832, 848, 849
<code>\gls@debug@nr</code> 7, 375, 393, 624, 643, 878		<code>\gls@inlinesubsep</code>	274, 275, 579, 595, 596, 832, 848, 849
<code>\gls@debug@val</code>	7, 375, 393, 624, 643, 878	<code>\gls@islistofacronyms</code>	21, 382, 383, 632, 886, 887
<code>\gls@defdocnewglossaryentry</code>	83, 85, 108, 417, 418, 432, 668, 669, 683, 924, 925, 940	<code>\gls@istfilebase</code>	47, 48, 48, 190, 191, 398, 480, 481, 648, 732, 733, 903, 904, 986, 987
<code>\gls@defglossaryentry</code>	83–85, 94, 95, 417, 418, 424, 668, 669, 675, 924, 925, 931	<code>\gls@kernelglossredefs@nr</code>	39, 898
<code>\gls@delimN</code>	237, 237, 1014	<code>\gls@kernelglossredefs@val</code>	39, 898
<code>\gls@delimR</code>	237, 237, 1014	<code>\gls@label</code>	240, 509, 761, 1015
<code>\gls@disablepagerefexpansion</code>	202, 202, 207, 487–489, 739–741, 994, 995, 997	<code>\gls@level</code>	81, 97, 98, 223, 416, 426, 499, 500, 666, 677, 751, 922, 933, 1007
<code>\gls@do@addxdyattribute</code>	56, 402, 652, 908	<code>\gls@local@stepcounter</code> 104, 104, 937	
<code>\gls@docdef@addlabel</code>	84, 84, 85, 668, 669, 924, 925	<code>\gls@measuring@reset</code>	104, 937, 938
<code>\gls@docdef@labellist</code> 84, 84, 668, 924		<code>\gls@mfirstuc@nr</code>	41, 899, 900
<code>\gls@docclearpage</code>	53, 54, 400, 401, 650, 651, 906	<code>\gls@mfirstuc@val</code>	41, 899
<code>\gls@doentryfmt</code>	71, 411, 661, 917	<code>\gls@newacronym</code>	240, 240, 1016
<code>\gls@dostubst</code>	127, 446, 696, 697, 950	<code>\gls@noidx@nonumberlist</code>	78, 198, 920, 993
<code>\gls@dotoc@title</code> 214, 226, 493, 501, 745, 753, 1000, 1001, 1008, 1009		<code>\gls@noidx@glossary</code>	196, 199, 483, 486, 735, 738, 990, 993
<code>\gls@end@sanitizesort</code>	25, 26, 385, 634, 635, 889	<code>\gls@nonumberlist@nr</code> 78, 414, 664, 920	
<code>\gls@endcheck</code>	82, 417, 667, 923	<code>\gls@nonumberlist@val</code>	78, 414, 664, 920
<code>\gls@glossary</code> 39, 40, 200, 208–210, 393, 487, 490, 491, 643, 739, 742, 743, 898, 899, 994, 997, 998		<code>\gls@nosetquote</code>	95, 184, 185, 188, 424, 475, 476, 479, 675, 726, 728, 731, 931, 981, 983, 985
<code>\gls@glsdefs@addlabel</code>	84, 85, 668, 669, 924, 925	<code>\gls@number</code>	207, 489, 741, 996
<code>\gls@glsdefs@labellist</code>	84, 84, 668, 924	<code>\gls@numberedsection@nr</code> 10, 226, 377, 501, 502, 627, 753, 881, 1009	
<code>\gls@gobbleopt</code>	72, 73, 411, 412, 661, 662, 917, 918	<code>\gls@numberedsection@val</code>	10, 226, 377, 501, 627, 753, 881, 1009
<code>\gls@grplabel</code>	270, 594, 847, 1031	<code>\gls@numberpage</code>	202, 203, 207, 488, 489, 739–741, 995, 996
<code>\gls@hypergroup@rerun</code>	272, 272, 594, 847, 1031	<code>\gls@org@glossaryentryfield</code>	215, 493, 745, 1001
<code>\gls@if@measuring</code> 103, 104, 937, 938		<code>\gls@org@glossarysubentryfield</code>	215, 493, 745, 1001
		<code>\gls@org@insert</code>	526, 528, 530, 778, 780, 781, 783
		<code>\gls@org@label</code>	103, 104, 937, 938
		<code>\gls@org@link</code>	104, 937, 938

<code>\gls@org@refstepcounter</code>	<code>\gls@tr@set@numbers@toctitle</code> ..
. 103 , 103 , 104 , 937 , 938 40 , 394 , 644 , 899
<code>\gls@org@target</code> 104 , 937 , 938	<code>\gls@tr@set@symbols@toctitle</code> ..
<code>\gls@org@Alph</code> 40 , 393 , 644 , 899
. 206 , 207 , 489 , 741 , 996 , 997	<code>\gls@translate@nr</code> . 30 , 388 , 638 , 892
<code>\gls@org@alph</code>	<code>\gls@translate@val</code> . 30 , 388 , 638 , 892
. 206 , 207 , 489 , 741 , 996 , 997	<code>\gls@wrglossary</code> 201 , 202 , 487 , 739 , 994
<code>\gls@org@arabic</code> 206 , 207 , 489 , 741 , 996	<code>\gls@xdy@set@lang</code> 33 ,
<code>\gls@org@number</code> 206 , 207 , 489 , 741 , 996	190 , 191 , 216 , 894 , 986 , 987 , 1001
<code>\gls@org@Roman</code>	<code>\gls@xdystring</code>
. 206 , 207 , 489 , 741 , 996 , 997	126 , 127 , 445 , 446 , 696 , 697 , 950
<code>\gls@org@romannumeral</code>	<code>\gls@xindy@glsnumbersfalse</code>
. 206 , 207 , 489 , 741 , 996 34 , 390 , 640 , 895
<code>\gls@org@the</code> 206 , 207 , 489 , 741 , 996	<code>\gls@xindy@glsnumberstrue</code>
<code>\gls@original@glossary</code> 33 , 390 , 640 , 894
. 38 , 39 , 392 , 393 , 642 , 643 , 898	<code>\gls@xr@key</code> 9 , 77 , 77 , 78 ,
<code>\gls@original@makeglossary</code> . 38 ,	376 , 377 , 414 , 626 , 664 , 880 , 920
40 , 392 , 393 , 642 , 643 , 898 , 899	<code>\gls@accessdisplay</code> 347 , 546 , 799
<code>\gls@protected@pagefmts</code>	<code>\gls@accessibility</code>
. 127 , 202 , 202 , 203 , 335 , 344 , 538 , 543 , 791 , 796
445 , 488 , 696 , 739 , 740 , 950 , 995	<code>\gls@saccsupp</code> 343 , 344 , 344 , 543 , 796 , 797
<code>\gls@Romanpage</code> 202 , 203 ,	<code>\gls@sacronymtrue</code> 20 , 382 , 632 , 886
207 , 488 , 489 , 739 – 741 , 995 , 997	<code>\gls@sacrpluralsuffix</code>
<code>\gls@romanpage</code> 202 , 203 , 45 , 45 , 241 , 250 , 251 , 255 –
207 , 488 , 489 , 739 – 741 , 995 , 996	257 , 396 , 509 , 515 , 518 – 520 ,
<code>\gls@save@numberlist</code> 11 , 12 , 211 ,	522 , 646 , 761 , 767 , 770 – 772 ,
377 , 378 , 492 , 627 , 743 , 881 , 999	775 , 902 , 1016 , 1022 , 1025 – 1027
<code>\gls@seen@index@nr</code> 9 , 376 , 626 , 880	<code>\gls@sacrshortcutsfalse</code>
<code>\gls@seen@index@val</code> 9 , 376 , 626 , 880 42 , 395 , 645 , 900
<code>\gls@set@xr@key</code> 77 , 77 , 414 , 664 , 920	<code>\gls@sacrshortcutstrue</code> 42 , 394 , 645 , 900
<code>\gls@start@measuring</code>	<code>\gls@sacspace</code> 249 , 249 , 251 ,
. 104 , 104 , 105 , 937 , 938	513 , 514 , 516 , 765 – 767 , 1020 – 1022
<code>\gls@stop@measuring</code> 104 , 104 , 105 , 938	<code>\gls@sadd</code> 39 , 176 , 176 , 177 ,
<code>\gls@suffixF</code> 49 , 49 , 183 , 185 ,	393 , 470 , 471 , 643 , 722 , 898 , 977
398 , 474 , 476 , 576 , 578 , 648 ,	<code>\gls@saddall</code> 177 , 471 , 722 , 977
726 – 728 , 830 , 831 , 904 , 980 – 982	<code>\gls@saddallunused</code> . 177 , 471 , 722 , 977
<code>\gls@suffixFF</code> 49 , 49 , 183 , 185 ,	<code>\gls@saddkey</code> 88 , 420 , 671 , 927
398 , 474 , 476 , 576 – 578 , 648 ,	<code>\GlsAddLetterGroup</code> . 63 , 407 , 657 , 913
726 , 728 , 830 , 831 , 904 , 980 – 982	<code>\gls@saddprotectedpagefmt</code>
<code>\gls@text</code> 118 , 119 , 441 , 692 , 946 203 , 204 , 488 , 740 , 995
<code>\gls@the</code> 207 , 489 , 741 , 996	<code>\GlsAddSortRule</code> 61 , 405 , 406 , 656 , 911
<code>\gls@thissty</code> 31 , 389 , 639 , 893	<code>\gls@saddstoragekey</code> . 87 , 420 , 670 , 926
<code>\gls@tmp</code> 192 , 195 , 200 , 262 , 486 ,	<code>\GlsAddXdyAlphabet</code> . 57 , 403 , 653 , 909
530 , 738 , 783 , 988 , 990 , 993 , 1028	<code>\GlsAddXdyAttribute</code>
<code>\gls@tmplen</code> 134 , 322 , 324 , 451 , 585 , 55 , 56 , 56 , 57 , 402 , 403 ,
621 – 623 , 703 , 838 , 875 , 876 , 957	573 , 652 , 653 , 826 , 827 , 907 – 909
<code>\gls@tr@set@acronym@toctitle</code> . .	<code>\GlsAddXdyCounters</code> 55 , 55 ,
. 20 , 382 , 631 , 632 , 886	56 , 57 , 74 , 401 , 402 , 412 , 573 ,
<code>\gls@tr@set@main@toctitle</code>	651 – 653 , 662 , 826 , 907 , 908 , 918
. 19 , 381 , 631 , 885	

<code>\GlsAddXdyLocation</code>	<code>\GlsDeclareNoHyperList</code>
404, 405, 573, 655, 827, 910, 911 22, 23, 383, 633, 887
<code>\GlsAddXdyStyle</code>	<code>\glsdefaultshortaccess</code>
61, 406, 656, 912 371, 371, 567–570, 820–823
<code>\glsautomakefalse</code> 36, 37, 193, 391,	<code>\glsdefaultttype</code>
392, 482, 642, 734, 897, 988, 989 19, 19, 20, 33, 50, 62, 64,
<code>\glsautomaketrue</code> 36, 37, 391, 642, 897	71, 72, 96, 113, 200, 212, 213,
<code>\glsautoprefix</code>	381, 399, 406, 407, 411, 425,
10, 11,	435, 441, 486, 492, 493, 631,
227, 377, 502, 627, 754, 881, 1009	649, 656, 657, 661, 676, 686,
<code>\glsbackslash</code>	692, 738, 744, 885, 894, 904,
178, 471, 723, 977	912, 913, 917, 932, 943, 993, 1000
<code>\glscapitalisewords</code>	<code>\glsdefmain</code>
. 170, 170, 719, 973, 974 19,
<code>\glscapscase</code>	20, 74, 381, 412, 631, 662, 885, 918
114–117,	<code>\glsdefs@newdocentry</code> 85, 86, 669, 925
136–141, 157–164, 253, 254,	<code>\GLSdesc</code>
348–355, 365, 366, 435, 436,	147, 457, 709, 962
438–440, 452–455, 461–465,	<code>\Glsdesc</code>
517, 526, 530, 546, 547, 549–	147, 147, 457, 709, 962
554, 562, 686, 687, 689–691,	<code>\glsdesc</code>
704–706, 712–717, 769, 779,	146, 147, 457, 709, 962
783, 799, 800, 802–808, 815,	<code>\GLSdescplural</code>
943–945, 957–960, 966–971, 1024	148, 148, 457, 709, 963
<code>\glscategory</code>	<code>\glsdescplural</code>
343, 543, 796 147, 148, 457, 709, 962, 963
<code>\glsclearpage</code>	<code>\glsdescriptionaccessdisplay</code> . .
. 53, 54, 400, 401, 650, 651, 906 345, 357,
<code>\glsclosebrace</code>	361, 544, 547–549, 556, 558,
. 60, 178, 182, 183, 405, 471,	559, 798, 801, 802, 809, 811, 812
474, 576, 577, 655, 723, 726,	<code>\glsdescriptionpluralaccessdisplay</code>
829, 830, 910, 911, 977, 980, 981 346, 357,
<code>\glscompositor</code> 48, 48, 49, 59, 185,	544, 546, 547, 556, 798–800, 809
398, 404, 476, 576, 578, 648,	<code>\glsdescwidth</code>
654, 727, 829, 831, 904, 910, 982	280, 280, 281–
<code>\glscounter</code>	283, 285, 286, 288–291, 291,
22,	292–296, 303, 303, 304–306,
22, 43, 55, 73, 98, 124, 383,	308–310, 310, 311–316, 598–
395, 401, 412, 427, 444, 573,	601, 603–607, 611–618, 852,
633, 645, 651, 662, 678, 695,	853, 855–861, 864–866, 868–872
826, 887, 901, 907, 918, 934, 948	<code>\glsdetoklabel</code>
<code>\glscurrententrylabel</code>	35, 65, 65, 66–70, 78, 79, 86,
211,	91–95, 101, 106–110, 123, 165,
215, 492, 493, 744, 745, 999, 1001	167, 175, 176, 196–198, 205,
<code>\glscurrentfieldvalue</code>	207, 215, 218, 219, 223, 224,
69,	226, 229, 231, 263–268, 322,
70, 70, 410, 411, 660, 661, 916	328–330, 344–347, 372, 373,
<code>\glscustomtext</code>	407–410, 415, 418, 422–424,
. 114, 116, 118, 136–141,	429, 431–433, 443, 465, 466,
157–164, 253, 254, 348, 350,	469, 470, 483–485, 489, 493,
352–355, 365, 366, 435, 438–	495, 496, 499–501, 503, 504,
440, 452–455, 461–465, 517,	535–538, 544–546, 571, 572,
522, 523, 526, 528, 530, 546,	590, 622, 641, 658–660, 665,
549, 551–554, 562, 686, 688–	669, 673–675, 679, 682–684,
691, 704–706, 712–717, 769,	694, 717, 721, 722, 735–737,
775, 778–781, 783, 799, 802,	
804, 806–808, 815, 816, 943–	
945, 957–960, 966–971, 1023, 1024	

740, 741, 745, 747, 748, 751–
753, 755, 756, 788–791, 797–
799, 824, 825, 843, 844, 875,
896, 913–916, 920, 921, 925,
929–931, 936, 939–941, 948,
971, 972, 976, 977, 991, 992,
996, 997, 1001, 1003, 1004,
1007, 1008, 1010, 1011, 1029, 1030
\glsdisablehyper . 135, 452, 703, 957
\Glsdisp 141, 141, 960
\glsdisp 140, 141, 455, 706, 960
\glsdisplay 119, 435, 441, 442, 686, 692
\glsdisplayfirst
. 119, 435, 441, 686, 692
\glsdisplaynumberlist 174,
197, 469, 484, 721, 736, 976, 992
\glsdohyperlink
134, 135, 451, 452, 703, 704, 957
\glsdohypertarget
134, 135, 451, 452, 703, 704, 957
\glsdoifexists 65, 67, 69,
91–94, 105, 106, 136–141, 157–
164, 170, 174–176, 197, 198,
330–334, 353–355, 360, 361,
408–410, 422–424, 430, 431,
452–455, 461–465, 467, 469,
470, 484, 485, 553, 554, 558,
591–593, 658–660, 673–675,
681, 682, 704–707, 712–716,
719, 721, 722, 736, 737, 806,
807, 811, 812, 844–847, 913–
916, 929–931, 938, 939, 957–
960, 966–971, 974–977, 991, 992
\glsdoifexistsordo
. 66, 122, 165, 408, 443,
465, 658, 693, 717, 914, 947, 971
\glsdoifexistsorwarn
. 66, 225, 231, 232,
408, 501, 504, 505, 658, 753,
756, 757, 914, 1008, 1011, 1012
\glsdoifnoexists
. 66, 83, 94, 408, 417,
424, 658, 667, 675, 914, 923, 931
\glsdonohyperlink
. 124, 134, 134, 135,
444, 451, 452, 695, 703, 948, 957
\glsdoparenifnotempty
. 261, 530, 782, 1028
\glsdosanitizesort
. 16, 379, 380, 629, 883
\glsdoshowtarget . . . 8, 134, 880, 957
\glsenableentrycount
108, 432, 433, 683, 684, 940, 941
\glsenablehyper . . 135, 452, 703, 957
\glsencapwrcontent 124, 124, 948
\glsentryaccess
. . . . 340, 345, 542, 544, 795, 797
\glsentrycounter
. . . . 124, 235, 238, 444, 506,
508, 695, 758, 760, 948, 1013, 1015
\glsentrycounterfalse
. 14, 379, 629, 883
\glsentrycounterlabel
230, 230, 503, 504, 755, 756, 1011
\GlsEntryCounterLabelPrefix 227,
229, 502, 503, 754, 755, 1009–1011
\glsentrycountertrue 14, 379, 629, 883
\glsentrycurrcount 108,
110, 432–434, 683–685, 940–942
\Glsentrydesc
147, 168, 168, 232, 357, 361,
457, 466, 505, 556, 558, 709,
718, 757, 809, 811, 962, 972, 1012
\glsentrydesc 147, 168,
168, 232, 357, 361, 436–438,
457, 466, 505, 547–549, 556,
558, 687–689, 709, 718, 756,
801, 802, 809, 811, 962, 972, 1012
\glsentrydescaccess
. . . . 341, 346, 542, 544, 795, 798
\Glsentrydescplural
. . . . 148, 168, 168, 357, 458,
467, 556, 709, 718, 809, 963, 973
\glsentrydescplural . . . 148, 168,
168, 357, 435, 436, 457, 458,
466, 546, 547, 556, 686, 687,
709, 718, 799, 800, 809, 963, 973
\glsentrydescpluralaccess
. . . . 341, 346, 542, 545, 795, 798
\Glsentryfirst 112, 115,
143, 169, 170, 349, 356, 434,
437, 439, 456, 467, 548, 550,
555, 685, 688, 690, 707, 719,
801, 804, 808, 942, 944, 961, 973
\glsentryfirst 111,
115, 116, 143, 169, 170, 349,
350, 356, 434, 436, 437, 439,
456, 467, 548, 550, 551, 555,
685, 687, 688, 690, 707, 719,
801–804, 808, 942, 944, 961, 973

<code>\glsentryfirstaccess</code>	355, 362, 366, 367, 434, 463, 466, 469, 513, 517, 518, 553,
. . . . 341 , 345 , 542 , 544 , 795 , 797	
<code>\Glsentryfirstplural</code>	554, 559, 562, 563, 685, 715, 717, 720, 721, 765, 769, 770,
114, 145 , 170 , 170 , 348 , 356 ,	806, 807, 812, 815, 816, 942, 969, 972, 975, 1020, 1024, 1025
435 , 436 , 438 , 456 , 467 , 547 ,	
550 , 555 , 686 , 687 , 689 , 708 ,	
719 , 800 , 803 , 808 , 943 , 962 , 973	
<code>\glsentryfirstplural</code>	<code>\glsentrylong</code>
. . . . 112 , 114 , 115 , 145 , 170 , 111 , 118 , 161 , 162 , 167 ,
170 , 348 , 349 , 356 , 434 – 436 ,	173 , 173 , 174 , 248 – 258 , 352 ,
438 , 456 , 457 , 467 , 546 , 547 ,	354 , 355 , 362 – 371 , 434 , 441 ,
549 , 550 , 555 , 685 – 687 , 689 ,	463 , 464 , 466 , 469 , 513 – 521 ,
708 , 719 , 799 , 800 , 802 , 803 ,	528 , 552 , 554 , 559 – 566 , 685 ,
808 , 809 , 942 , 943 , 961 , 962 , 973	691 , 715 , 717 , 720 , 721 , 765 – 773 , 781 , 806 – 808 , 812 – 820 ,
<code>\glsentryfirstpluralaccess</code>	942 , 946 , 969 , 972 , 975 , 1020 – 1027
. . . . 341 , 345 , 542 , 544 , 795 , 797	<code>\glsentrylongaccess</code>
<code>\glsentryfmt</code> 342 , 346 , 542 , 545 , 795 , 798
73 , 74 , 114 ,	<code>\Glsentrylongpl</code>
412 , 435 , 662 , 663 , 686 , 918 , 943	113 , 163 , 173 , 173 , 174 , 248 ,
<code>\Glsentryfull</code>	249 , 253 – 255 , 353 , 362 , 365 ,
. . . . 174 , 174 , 246 , 255 , 256 ,	367 , 435 , 465 , 469 , 513 , 514 ,
367 , 369 , 469 , 512 , 518 , 519 ,	517 , 518 , 553 , 559 , 562 , 563 ,
563 , 565 , 721 , 764 , 770 , 771 ,	686 , 716 , 720 , 721 , 765 , 766 ,
816 , 818 , 975 , 1018 , 1025 , 1026	769 , 770 , 806 , 812 , 815 – 817 ,
<code>\glsentryfull</code>	943 , 970 , 975 , 1020 , 1024 , 1025
. . . . 174 , 174 , 246 , 255 , 256 ,	<code>\glsentrylongpl</code>
367 , 369 , 469 , 512 , 518 , 519 , 112 , 118 , 163 , 164 , 173 ,
563 , 565 , 720 , 764 , 770 , 771 ,	173 , 174 , 248 – 250 , 253 – 256 ,
816 , 818 , 975 , 1018 , 1025 , 1026	353 , 362 , 363 , 365 , 367 – 369 ,
<code>\Glsentryfullpl</code>	434 , 441 , 464 , 465 , 469 , 513 ,
. . . . 174 , 174 , 246 , 255 , 256 ,	514 , 517 – 520 , 528 , 534 , 553 ,
367 , 369 , 469 , 512 , 518 , 520 ,	559 , 560 , 562 – 565 , 685 , 692 ,
563 , 565 , 721 , 764 , 770 , 771 ,	716 , 717 , 720 , 721 , 765 , 766 ,
817 , 818 , 975 , 1019 , 1025 , 1026	769 – 771 , 781 , 787 , 806 , 812 ,
<code>\glsentryfullpl</code>	813 , 815 – 818 , 942 , 946 , 970 ,
. . . . 174 , 174 , 246 , 255 , 256 ,	971 , 975 , 1020 , 1021 , 1024 – 1026
367 , 369 , 469 , 512 , 518 , 520 ,	<code>\glsentrylongpluralaccess</code>
563 , 565 , 721 , 764 , 770 , 771 , 342 , 346 , 542 , 545 , 796 , 798
816 , 818 , 975 , 1019 , 1025 , 1026	<code>\Glsentryname</code>
<code>\glsentryitem</code>	146,
230 ,	166, 166 , 231 , 357 , 361 , 457 ,
274 , 276 – 282 , 284 , 292 , 293 ,	466 , 504 , 556 , 558 , 708 , 717 ,
295 , 303 , 305 , 307 , 310 , 312 ,	756 , 809 , 811 , 962 , 971 , 972 , 1012
314 , 318 , 319 , 321 , 323 , 504 ,	<code>\glsentryname</code>
579 – 584 , 586 – 588 , 595 , 597 – 600 , 604 – 606 , 611 – 613 , 615 – 617 , 619 – 622 , 756 , 832 – 841 ,	146, 166 , 166 , 211 ,
848 , 850 – 854 , 858 – 860 , 865 – 867 , 869 – 871 , 873 , 874 , 876 , 1011	322 , 357 , 360 , 457 , 466 , 555 ,
<code>\Glsentrylong</code>	556 , 558 , 622 , 708 , 709 , 717 ,
. . . . 112 , 161 , 167 , 173 , 173 ,	809 , 811 , 875 , 962 , 971 , 972 , 999
174 , 248 , 249 , 254 , 255 , 353 ,	<code>\glsentrynumberlist</code> 174 , 174 , 197 ,
	469 , 484 , 721 , 736 , 975 , 976 , 991
	<code>\glsentryparent</code>
 171 , 468 , 719 , 974
	<code>\Glsentryplural</code> 114 , 144 , 169 , 169 ,

348, 356, 436, 438, 456, 467, 546, 549, 555, 686, 689, 708, 718, 800, 803, 808, 943, 961, 973	\Glsentryshortpl . 116, 159, 173, <u>173</u> , 250, 256, 351, 363, 368, 369, 439, 462, 469, 514, 519, 520, 551, 560, 564, 565, 690, 714, 720, 766, 771, 804, 813, 817, 818, 944, 968, 975, 1021, 1026
\glsentryplural 114, 115, 144, <u>168</u> , 169, 348, 349, 356, 435, 436, 438, 456, 467, 546, 547, 549, 550, 555, 686, 687, 689, 708, 718, 799, 800, 802, 803, 808, 943, 961, 973	\glsentryshortpl 116, 118, 159, 160, <u>173</u> , <u>173</u> , 174, 248–250, 254– 256, 350, 351, 353, 362, 363, 367–369, 439, 441, 462, 463, 469, 513, 514, 518–520, 534, 551, 553, 559, 560, 563–565, 690, 692, 714, 720, 721, 765, 766, 770, 771, 787, 804, 806, 812, 813, 816–818, 944, 946, 968, 975, 1020, 1021, 1025, 1026
\glsentrypluralaccess <u>341</u> , 345, 542, 544, 795, 797	\glsentryshortpluralaccess <u>341</u> , 346, 542, 545, 795, 798
\Glsentryprefix 329, <u>329</u> , 332, 590, 592, 843, 845	\glsentrysort <u>171</u> , 468, 719, 974
\glsentryprefix . . <u>328</u> , 329, 330, 333, 590, 591, 593, 843, 844, 846	\Glsentrysymbol 149, 169, <u>169</u> , 232, 358, 361, 458, 467, 505, 556, 559, 710, 718, 757, 809, 812, 963, 973, 1012
\Glsentryprefixfirst <u>328</u> , 329, 332, 590, 592, 843, 845	\glsentrysymbol . . 149, 169, <u>169</u> , 232, 358, 361, 436–438, 458, 467, 505, 523, 526, 530, 547– 549, 556, 558, 687–689, 709, 710, 718, 757, 775, 779, 783, 801, 802, 809, 812, 963, 973, 1012
\glsentryprefixfirst . <u>328</u> , 329, 330, 333, 590, 591, 593, 843, 844, 846	\glsentrysymbolaccess <u>341</u> , 345, 542, 544, 795, 797
\Glsentryprefixfirstplural 329, <u>329</u> , 333, 590, 593, 843, 846	\Glsentrysymbolplural 150, 169, <u>169</u> , 358, 458, 467, 556, 710, 719, 810, 964, 973
\glsentryprefixfirstplural <u>328</u> , 329, 331, 334, 590, 591, 594, 843, 845, 847	\glsentrysymbolplural . 150, 169, <u>169</u> , 358, 435, 436, 458, 467, 523, 526, 530, 546, 547, 556, 686, 687, 710, 719, 775, 779, 783, 799, 800, 810, 963, 964, 973
\Glsentryprefixplural 329, <u>329</u> , 332, 590, 592, 843, 846	\glsentrysymbolpluralaccess <u>341</u> , 345, 542, 544, 795, 798
\glsentryprefixplural <u>328</u> , 329, 331, 334, 590, 591, 593, 843, 845, 847	\Glsentrytext . 115, 142, 168, <u>168</u> , 349, 355, 437, 439, 455, 467, 548, 550, 555, 688, 690, 707, 718, 801, 803, 808, 944, 960, 973
\glsentryprevcount 108– 110, 432, 433, 683, 684, 940, 941	\glsentrytext 115, 116, 142, 168, <u>168</u> , 176, 211, 349, 350, 355, 436, 437, 439, 455, 467, 470, 492, 547, 548, 550, 551, 555, 687–690,
\Glsentryshort 117, 157, 167, 173, <u>173</u> , 250, 256, 351, 353, 363, 368, 369, 440, 461, 466, 468, 514, 519, 552, 553, 560, 564, 565, 691, 713, 718, 720, 766, 771, 805, 806, 813, 817, 818, 945, 967, 972, 975, 1021, 1025, 1026	
\glsentryshort 117, 118, 157, 158, 167, <u>172</u> , 173, 174, 247–258, 351–354, 362–371, 440, 441, 461, 462, 466, 468, 469, 512– 521, 551–554, 559–566, 691, 712, 713, 718, 720, 721, 764– 773, 805–807, 812–820, 945, 946, 966, 967, 972, 975, 1019–1027	
\glsentryshortaccess <u>341</u> , 346, 542, 545, 795, 798	

707, 718, 722, 743, 801, 803,
 804, 808, 944, 960, 973, 976, 999
 \glstrytextaccess
 340, 345, 542, 544, 795, 797
 \glstrytitlecase
 170, 467, 468, 719, 974
 \glstrytype 96,
 170, 425, 468, 676, 719, 932, 974
 \Glsentryuseri
 151, 171, 171, 358, 459,
 468, 557, 710, 720, 810, 964, 974
 \glstryuseri
 151, 171, 171, 358, 458, 459,
 468, 557, 710, 719, 810, 964, 974
 \glstryuseriaccess
 342, 346, 542, 545, 796, 798
 \Glsentryuserii
 152, 171, 171, 359, 459,
 468, 557, 711, 720, 810, 964, 974
 \glstryuserii
 152, 171, 171, 359, 459, 468,
 557, 710, 711, 720, 810, 964, 974
 \glstryuseriiaccess
 342, 347, 542, 545, 796, 798
 \Glsentryuseriii
 153, 171, 172, 359, 459,
 468, 557, 711, 720, 810, 965, 974
 \glstryuseriii
 153, 171, 172, 359, 459,
 468, 557, 711, 720, 810, 965, 974
 \glstryuseriiiaccess
 342, 347, 543, 545, 796, 798
 \Glsentryuseriv
 154, 172, 172, 359, 460,
 468, 557, 711, 720, 810, 965, 974
 \glstryuseriv
 154, 172, 172, 359, 460, 468,
 557, 711, 720, 810, 811, 965, 974
 \glstryuserivaccess
 342, 347, 543, 545, 796, 799
 \Glsentryuserv
 155, 172, 172, 360, 460,
 468, 557, 712, 720, 811, 966, 975
 \glstryuserv 155, 172,
 172, 360, 460, 468, 557, 558,
 712, 720, 811, 965, 966, 974, 975
 \glstryuservaccess
 342, 347, 543, 545, 796, 799
 \Glsentryuservi
 156, 172, 172, 360, 460,
 468, 558, 712, 720, 811, 966, 975
 \glstryuservi
 156, 172, 172, 360, 460, 461,
 468, 558, 712, 720, 811, 966, 975
 \glstryuserviaccess
 342, 347, 543, 546, 796, 799
 \glscslocationsfalse
 12, 196, 483, 735, 882, 990
 \glscslocationstrue 378, 627
 \glsexpandfields .. 83, 417, 667, 923
 \glffieldaccsupp
 343, 343, 344, 543, 544, 796, 797
 \glffielddef 92, 423, 673, 929
 \glffieldedef 91, 422, 673, 929
 \glffieldfetch 92,
 170, 423, 467, 674, 719, 930, 974
 \glffieldgdef 91, 422, 673, 929
 \glffieldxdef 91, 422, 673, 929
 \glfindwidesttoplevelname
 322, 621, 875
 \GLSfirst 143, 456, 707, 961
 \Glsfirst 143, 143, 456, 707, 961
 \glsfirst 143, 143, 456, 707, 960, 961
 \glsfirstaccessdisplay
 345, 349, 350, 356, 544, 548,
 550, 551, 555, 797, 801–804, 808
 \GLSfirstplural .. 145, 456, 708, 962
 \Glsfirstplural 145, 145, 456, 708, 961
 \glsfirstplural 144, 145, 456, 708, 961
 \glsfirstpluralaccessdisplay ..
 345, 348, 356,
 544, 546, 547, 549, 550, 555,
 797, 799, 800, 802, 803, 808, 809
 \glsfirstpluralacessdisplay ...
 349, 550, 803
 \glsgenacfmt 116, 248, 249,
 255, 350, 362, 368, 439, 513,
 514, 519, 551, 559, 560, 564,
 690, 765, 766, 770, 804, 812,
 813, 817, 944, 1019–1021, 1025
 \glsgenentryfmt
 114, 114, 248, 249, 254,
 255, 260, 348, 362, 366, 368,
 438, 513, 514, 517, 519, 521,
 523, 524, 526, 528, 530, 532,
 534, 549, 559, 560, 562, 564,
 689, 765, 766, 769, 770, 773,
 775, 777, 778, 781, 783, 785,
 787, 802, 812, 813, 816, 817,
 943, 1019–1021, 1024, 1025, 1028

<code>\glsgetgrouplabel</code>	234 , 506 , 758 , 1013	528 , 530 , 546 , 549 , 551–554 ,
<code>\glsgetgrouptitle</code> 234 , 273 , 277–279 ,	562 , 686 , 689–691 , 704–706 ,
	297–302 , 318–322 , 325 , 506 ,	712–717 , 769 , 775 , 778 , 781 ,
	595 , 597 , 598 , 608–611 , 619–	783 , 799 , 802 , 804–808 , 815 ,
	621 , 623 , 758 , 848 , 850 , 851 ,	943–945 , 957–960 , 966–971 , 1024
	861–864 , 873–875 , 877 , 1012 , 1032	<code>\glsifusedtranslatordict</code>
<code>\gls glossarymark</code>	51 , 51 , 399 , 649 , 905 29 , 388 , 638 , 892
<code>\gls groupheading</code> 183 , 224 ,	<code>\glsifusetranslator</code>
	233 , 274 , 277–279 , 281 , 282 ,	29–
	284 , 292 , 293 , 295 , 297–303 ,	31 , 46 , 47 , 326 , 388 , 389 , 396 ,
	305 , 307 , 310 , 312 , 314 , 317–	397 , 572 , 589 , 637–639 , 646 ,
	323 , 325 , 474 , 500 , 506 , 577 ,	647 , 825 , 842 , 892 , 893 , 902 , 903
	595 , 597–600 , 604–606 , 608–	<code>\glsignore</code>
	613 , 615–617 , 619–623 , 726 ,	177 , 471 , 722 , 777
	752 , 758 , 830 , 848 , 850–854 ,	<code>\glsindexingsetting</code>
	858–867 , 869–877 , 981 , 1007 , 1012	18 , 193 , 195 , 883 , 885 , 989 , 990
<code>\gls groupskip</code>	<code>\glsindexonlyfirstfalse</code>
 182 , 183 , 224 , 233 , 275 , 31 , 389 , 639 , 893
	277 , 281 , 282 , 284 , 287 , 288 ,	<code>\glsinlinedescformat</code>
	292 , 294 , 295 , 304 , 305 , 307 , 274 , 275 , 579 , 596 , 832 , 849
	311 , 313 , 315 , 318 , 320 , 321 ,	<code>\glsinlinedopostchild</code>
	324 , 474 , 500 , 506 , 576 , 577 , 274 , 275 , 579 , 596 , 832 , 849
	596 , 597 , 599 , 600 , 602 , 605 ,	<code>\glsinlineemptydescformat</code>
	606 , 611–613 , 615–617 , 619– 274 , 275 , 579 , 596 , 832 , 849
	621 , 623 , 726 , 752 , 758 , 829 ,	<code>\glsinlineifhaschildren</code> ..
	830 , 849 , 850 , 852–856 , 858–	274 , 275
	860 , 865–867 , 869–871 , 873–	<code>\glsinlinenameformat</code>
	875 , 877 , 980 , 981 , 1007 , 1012 274 , 275 , 579 , 596 , 832 , 849
<code>\gls hyperfirstfalse</code>	<code>\glsinlineparentchildseparator</code>
	255 , 368 , 519 , 564 , 771 , 817 , 1025 274 , 275 , 579 , 596 , 832 , 849
<code>\gls hyperfirsttrue</code>	.. 31 , 389 , 639 , 893	<code>\glsinlinesubdescformat</code>
<code>\gls hyperlink</code> 175 , 275 , 276 , 579 , 596 , 832 , 849 , 850
	211 , 470 , 492 , 722 , 743 , 976 , 999	<code>\glsinlinesubnameformat</code>
<code>\gls hypernavsep</code> 275 , 276 , 579 , 596 , 832 , 849 , 850
 273 , 273 , 595 , 848 , 1032	<code>\glsinlinesubseparator</code>
<code>\gls hypernumber</code> 275 , 275 , 579 , 596 , 832 , 849
 50 , 237 , 239 , 398 , 507–	<code>\glsinsert</code>
	509 , 649 , 759–761 , 904 , 1014 , 1015	114–118 , 136–141 , 157–
<code>\gls ifhyper</code> 121 , 442 , 693	164 , 253 , 254 , 348–355 , 365 ,
<code>\gls ifhyperon</code>	366 , 435–440 , 452–455 , 461–
 123 , 442 , 443 , 693 , 694 , 948	465 , 517 , 523 , 526 , 528 , 530 ,
<code>\gls IfListOfAcronyms</code> 21 ,	546–554 , 562 , 686–691 , 704–
	21 , 22 , 382 , 383 , 632 , 886 , 887	706 , 712–717 , 769 , 775 , 778–
<code>\gls ifmeasuring</code>	.. 104 , 104 , 937 , 938	781 , 783 , 799–808 , 815 , 816 ,
<code>\gls ifplural</code>	943–945 , 958–960 , 966–971 , 1024
	114 , 116 , 117 , 136–141 , 157–	<code>\glskeylisttok</code>
	164 , 253 , 348 , 350 , 351 , 353– 245 , 245 , 246 , 371 , 511 , 522–
	355 , 365 , 435 , 438–440 , 452–	525 , 527 , 529 , 531–535 , 567–
	455 , 461–465 , 517 , 523 , 526 ,	571 , 763 , 773 , 774 , 776–782 ,
		784–786 , 788 , 820–824 , 1017 , 1018

`\glslabel` .. 95, 114–118, 122–124, 157–164, 248, 249, 253–255, 348–352, 362, 365, 366, 368, 425, 435–440, 443, 444, 461–465, 513, 514, 517, 519, 523, 526, 528, 530, 546–552, 559, 560, 562, 564, 675, 686–691, 694, 695, 713–717, 765, 766, 769, 770, 775, 778, 779, 781, 783, 799–805, 812, 813, 815, 817, 931, 943–945, 947, 948, 967–971, 1019–1021, 1024, 1025
`\glslabeltok` .. 245, 245, 246, 511, 521–535, 567–570, 763, 773–783, 785–788, 820–823, 1017, 1018
`\glsletentryfield` 165, 465, 717, 971
`\Glslink` 122, 122, 947
`\glslink` .. 122, 122, 246, 254–256, 260, 366–369, 442, 511, 512, 518, 519, 522, 563–565, 693, 763, 764, 769–771, 775, 816, 818, 947, 1018, 1024–1026, 1028
`\glslinkcheckfirsthyperhook` ...
 123, 123, 443, 694, 947, 948
`\glslinkpostsetkeys`
 123, 124, 443, 444, 694, 948
`\glslinkvar` 121, 121, 442, 693, 946, 947
`\glslistdottedwidth`
 279, 279, 580, 598, 833, 851
`\glslistexpandedname` .. 276, 277, 850
`\glslistgroupheaderfmt`
 276, 277–279, 597, 598, 850, 851
`\glslistinit` 276, 277, 850
`\glslistnavigationitem`
 276, 278, 279, 597, 850, 851
`\glslocalreset` 105, 107, 430, 431, 681, 682, 938, 939
`\glslocalresetall` 107, 431, 682, 939
`\glslocalunset` 106, 107, 120, 431, 432, 452–455, 682, 683, 704–706, 939, 940, 946
`\glslocalunsetall` 107, 432, 683, 940
`\glslocationcstoencap` 50, 904
`\glslongaccessdisplay` . 346, 352–355, 362–371, 545, 552–554, 559–567, 798, 806–808, 812–820
`\glslongaccesskey` 371, 571, 824
`\glslongkey`
 241, 371, 509, 571, 761, 824, 1016
`\glslongpluralaccessdisplay` ...
 346, 353, 362, 363, 365, 367–369, 545, 553, 559, 560, 562–565, 567, 798, 806, 812, 813, 815–818, 820
`\glslongpluralaccesskey` 372, 571, 824
`\glslongpluralkey`
 241, 372, 509, 571, 761, 824, 1016
`\glslongtok` 245, 245, 246, 248–250, 254, 255, 362, 366, 368, 371, 511–514, 518, 519, 521–529, 531–535, 559, 560, 563, 564, 567–570, 763–766, 769, 771, 773–782, 784–788, 812, 813, 816, 817, 820–824, 1018–1021, 1024, 1025
`\glslowercase` 4, 877
`\glsLTpenaltycheck`
 290, 290, 291, 603, 604, 857
`\glsmakefirststuc` 41, 900
`\glsmcols` 297, 297, 298–302, 607–610, 861–864
`\glsmeasuredepth` 104, 938
`\glsmeasureheight` 104, 134, 938, 957
`\glsmeasurewidth` . 105, 322–324, 938
`\glsmfuaddmap` 5, 90, 111, 112, 122, 136, 139, 141–157, 159, 161, 163, 166, 168–174, 231, 232, 243, 244, 329, 331, 332, 877, 878, 928, 929, 942, 947, 958–966, 968–970, 972–975, 1012, 1017
`\glsmfublocker` 5, 5, 90, 137, 140, 142–156, 158, 160, 162, 164, 877, 878, 929, 958–970
`\glsmfuexcl` .. 5, 5, 176, 877, 878, 977
`\glsmoveentry` 101, 429, 679, 936
`\GLSname` 146, 457, 708, 962
`\Glsname` 146, 146, 457, 708, 962
`\glsname` 145, 146, 457, 708, 962
`\glsnameaccessdisplay`
 344, 357, 360, 361, 544, 555, 556, 558, 559, 797, 809, 811, 812
`\glsnamefont` 231, 236, 360, 361, 504, 505, 507, 538, 558, 756, 757, 759, 791, 811, 1011, 1012, 1014
`\glsnavhypergroupdotarget` . 272, 272
`\glsnavhyperlink` 270, 273, 594, 595, 847, 848, 1031, 1032
`\glsnavhyperlinkname`
 ... 270, 270, 272, 594, 847, 1031

<code>\glsnavhypertarget</code>	271 , 277–279 , 298–302 , 319 , 320 , 322 , 325 , 594 , 597 , 598 , 608– 611 , 620 , 621 , 623 , 847 , 850 , 851 , 861–864 , 873–875 , 877 , 1031	<code>\glsnumlistlastsep</code>	175 , 175 , 197 , 470 , 485 , 722 , 736 , 976 , 992	
<code>\glsnavigation</code> 273 , 277–279 , 298–302 , 319–321 , 325 , 594 , 597 , 608– 610 , 620 , 621 , 623 , 848 , 850 , 851 , 861–864 , 873–875 , 877 , 1031	<code>\glsnumlistparser</code>	175 , 194 , 470 , 483 , 722 , 734 , 976 , 990	
<code>\glsnavigationitem</code>	273 , 273	<code>\glsnumlistsep</code>	175 , 175 , 197 , 470 , 485 , 722 , 736 , 976 , 992	
<code>\glsnextpages</code>	12 , 78 , 214 , 229 , 378 , 414 , 493 , 503 , 627 , 664 , 745 , 755 , 882 , 920 , 1001 , 1010	<code>\glsopenbrace</code> 60 , 178 , 182 , 183 , 405 , 471 , 474 , 576 , 577 , 655 , 722 , 726 , 829 , 830 , 910 , 911 , 977 , 980 , 981	
<code>\glsnoexpandfields</code> .	83 , 417 , 667 , 923	<code>\glsorder</code> 32 , 32 , 190–192 , 194 , 220 , 390 , 480–482 , 497 , 639 , 640 , 731–734 , 749 , 894 , 986–989 , 1004	<code>\glsorg@endtheglossary</code> 6 , 6 , 374 , 624 , 878
<code>\glsnogroupskipfalse</code>	14 , 378 , 628 , 882	<code>\glsorg@PrintChanges</code>	6 , 374 , 624 , 878	
<code>\glsnoidxdisplayloc</code> 198 , 199 , 225 , 485 , 486 , 501 , 737 , 738 , 753 , 992 , 993 , 1008	<code>\glsorg@theglossary</code>	6 , 6 , 374 , 624 , 878	
<code>\glsnoidxdisplayloclisthandler</code> 198 , 225 , 485 , 500 , 737 , 752 , 992 , 1008	<code>\glspagelistwidth</code>	280 , 282 , 283 , 285 , 286 , 288–290 , 291 , 293–296 , 303 , 305 , 306 , 308 , 309 , 310 , 312–316 , 598– 601 , 603–607 , 611–618 , 852 , 853 , 855–861 , 865 , 866 , 868–872	
<code>\glsnoidxloclist</code>	197 , 224 , 224 , 484 , 500 , 736 , 751 , 752 , 991 , 1007 , 1008	<code>\glspar</code>	47 , 398 , 648 , 903	
<code>\glsnoidxloclisthandler</code> 224 , 225 , 500 , 752 , 1008	<code>\glspatchLTOoutput</code>	287–289 , 290 , 602 , 603 , 855–857	
<code>\glsnoidxnumberlistloophandler</code>	198 , 199 , 485 , 486 , 737 , 992 , 993	<code>\glspatchtabularx</code>	105 , 430 , 681 , 938	
<code>\glsnoidxprenumberlist</code> 78 , 223 , 224 , 920 , 1007	<code>\glspenaltygroupskip</code>	287 , 288 , 290 , 602 , 603 , 855–857	
<code>\glsnoidxstripaccents</code>	26 , 26 , 385 , 386 , 635 , 636 , 889 , 890	<code>\glspercentchar</code> 86 , 178 , 182 , 184 , 418 , 419 , 471 , 474 , 475 , 669 , 670 , 723 , 725 , 727 , 925 , 926 , 977 , 980 , 981	
<code>\glsnomakeindexwarning</code>	54 , 185 , 401 , 476 , 651 , 728 , 907 , 983	<code>\GLSpl</code>	140 , 454 , 706 , 959	
<code>\glsnonextpages</code> 78 , 214 , 228 , 414 , 493 , 503 , 664 , 745 , 755 , 920 , 1001 , 1010	<code>\Glspl</code>	113 , 139 , 139 , 259 , 435 , 454 , 521 , 685 , 705 , 773 , 943 , 959 , 1028	
<code>\glsnopostdotfalse</code> .	13 , 378 , 628 , 882	<code>\glspl</code>	112 , 138 , 139 , 259 , 434 , 453 , 521 , 685 , 705 , 773 , 942 , 958 , 959 , 1028	
<code>\glsnoxindywarning</code> . . .	49 , 54 , 55 , 57 , 58 , 60–63 , 178 , 179 , 398 , 401–403 , 405 , 406 , 471 , 472 , 648 , 651–653 , 655–657 , 723 , 724 , 904 , 907–909 , 911 , 912 , 978	<code>\GLSplural</code>	144 , 456 , 708 , 961	
<code>\glsnumberformat</code>	49 , 175 , 398 , 469 , 470 , 649 , 721 , 904 , 976	<code>\Glsplural</code> . . .	144 , 144 , 456 , 708 , 961	
<code>\glsnumberlistloop</code> 198 , 199 , 485 , 737 , 992 , 993	<code>\glsplural</code> . . .	143 , 144 , 456 , 708 , 961	
<code>\glsnumbersgroupname</code> .	40 , 44 , 47 , 235 , 394 , 396 , 397 , 506 , 644 , 646 , 648 , 758 , 899 , 902 , 903 , 1013	<code>\glspluralaccessdisplay</code> 345 , 348 , 349 , 356 , 544 , 546 , 547 , 549 , 550 , 555 , 797 , 799 , 800 , 802 , 803 , 808	
		<code>\glspluralsuffix</code>	45 , 45 , 98 , 246 , 248–250 , 362 , 363 , 367–370 , 396 , 427 , 513 , 514 , 559 , 560 , 564 , 566 , 646 ,	

677, 678, 765, 766, 813, 817–
 819, 902, 934, 1018, 1020, 1021
 \glspostdescription . 13, 47, 275,
 277, 278, 281, 292, 303, 310,
 311, 318, 319, 321, 323, 324,
 378, 398, 579–587, 596–599,
 604, 611, 615, 619–623, 628,
 648, 833, 835–841, 849–852,
 858, 865, 869, 873–876, 882, 903
 \glspostinline 274, 275, 595, 596, 848, 849
 \glspostlinkhook 122, 122,
 136–141, 157–164, 353–355,
 443, 452–455, 461–465, 553–
 555, 694, 704–707, 713–717,
 806–808, 947, 958–960, 966–971
 \glsprefixsep 330,
 330, 331–334, 591–594, 844–847
 \glsprestandardsort
 15, 16, 379, 380, 629, 883, 884
 \glsquote 178, 471, 723, 977
 \glsrefentry 229, 503, 755, 1011
 \glsreset 105,
 107, 430, 431, 681, 682, 938, 939
 \glsresetall 107, 431, 682, 939
 \glsresetcurrcountfalse 108, 940
 \glsresetentrycounter
 229, 503, 755, 1010
 \glsresetentrylist
 182, 222, 228, 228,
 474, 499, 502, 503, 576, 726,
 750, 754, 755, 829, 980, 1006, 1010
 \glsresetsubentrycounter
 229, 230, 274, 503, 504, 579,
 596, 755, 756, 832, 849, 1010, 1011
 \glsrestoreLToutput 290, 603, 857
 \glssanitizeortfalse
 28, 387, 637, 891
 \glssanitizeorttrue
 28, 387, 636, 637, 891
 \glsavenumberlistfalse
 12, 378, 627, 881
 \glsavewritesfalse 38, 392, 642, 898
 \glssee 210, 491, 743, 999
 \glsseeformat
 181, 196, 198, 210, 473,
 484, 485, 491, 576, 725, 735,
 737, 743, 829, 979, 991, 992, 999
 \glsseeitem 211, 211, 491, 492, 743, 999
 \glsseeitemformat
 211, 211, 492, 743, 999
 \glsseelastsep 211, 211, 491, 743, 999
 \glsseelist 210, 210, 491, 743, 999
 \glsseesep 211, 211, 491, 743, 999
 \glsentencecase
 4, 118, 119, 122, 141, 165–
 167, 877, 946, 947, 960, 971, 972
 \glsSetAlphaCompositor
 49, 398, 648, 904
 \glsSetCompositor 48, 398, 648, 904
 \glssetexpandfield
 24, 24, 27–29, 384, 386,
 387, 634, 636, 637, 888, 890, 891
 \glssetnoexpandfield
 24, 24, 25, 27, 28, 384, 386,
 387, 634, 636, 637, 888, 890, 891
 \GlsSetQuote 95, 184,
 185, 424, 475, 476, 479, 675,
 726, 728, 731, 931, 981, 983, 985
 \glsSetSuffixF 49, 398, 648, 904
 \glsSetSuffixFF 49, 398, 648, 904
 \glssettocitle
 44, 47, 214, 396, 397,
 493, 646, 647, 745, 901, 903, 1000
 \glssetwidest 322, 622, 875
 \GlsSetWriteIstHook 185, 476, 728, 983
 \GlsSetXdyCodePage
 62, 406, 656, 657, 912
 \GlsSetXdyFirstLetterAfterDigits
 178, 471, 723, 977, 978
 \GlsSetXdyLanguage 62, 406, 656, 912
 \GlsSetXdyLocationClassOrder . .
 61, 405, 655, 911
 \GlsSetXdyMinRangeLength
 179, 472, 723, 724, 978
 \GlsSetXdyNumberGroupOrder
 179, 471, 472, 723, 978
 \GlsSetXdyStyles 61, 406, 656, 912
 \glsshortaccessdisplay
 167, 346, 351–354, 362–371,
 466, 545, 551–554, 559–567,
 718, 798, 805–807, 812–820, 972
 \glsshortaccesskey 371, 571, 824
 \glsshortaccsupp 344, 344, 543, 796
 \glsshortkey
 241, 371, 509, 571, 761, 824, 1016
 \glsshortplaccsupp 344, 543, 796
 \glsshortpluralaccessdisplay . .
 346, 350, 351, 353, 362,

363, 367–369, 545, 551, 553,
 559, 560, 563–565, 567, 798,
 804, 806, 812, 813, 816–818, 820
 \glsshortpluralaccesskey
 371, 571, 824
 \glsshortpluralkey
 241, 371, 509, 571, 761, 824, 1016
 \glsshorttok 245, 245, 246, 371, 511,
 521–529, 531–535, 567–571,
 763, 773–788, 820–824, 1017, 1018
 \glsshowaccsupp 7,
 9, 375, 376, 625, 626, 879, 880
 \glsshowtarget
 ... 8, 8, 375, 376, 625, 879, 880
 \glsshowtargetfont
 8, 8, 375, 376, 625, 879
 \glsshowtargetfonttext ... 8, 8, 879
 \glsshowtargetinner 8, 8, 879
 \glsshowtargetouter 8, 8, 376, 625, 879
 \glsshowtargetsymbol
 8, 8, 376, 625, 879
 \glssortnumberfmt 16,
 17, 18, 380, 381, 629, 630, 884, 885
 \glsspace 242, 242, 510, 761, 774, 1016
 \glstepentry 229,
 230, 503, 504, 755, 756, 1010, 1011
 \glstepsubentry 229,
 230, 503, 504, 755, 756, 1010, 1011
 \glssubentrycounterfalse
 15, 379, 629, 883
 \glssubentrycounterlabel
 230, 230, 503, 504, 755, 756, 1011
 \glssubentryitem
 230, 275, 277–279, 281,
 282, 284, 292, 293, 295, 303,
 305, 307, 311, 312, 314, 318,
 319, 321, 323, 504, 579–588,
 596–600, 604–606, 611–613,
 615–617, 619–622, 756, 832–
 841, 849–854, 858–860, 865–
 867, 869–871, 873, 874, 876, 1011
 \GLSsymbol 149, 458, 710, 963
 \Glssymbol ... 149, 149, 458, 709, 963
 \glssymbol ... 148, 149, 458, 709, 963
 \glssymbolaccessdisplay .. 345,
 358, 361, 544, 547–549, 556,
 558, 559, 797, 801, 802, 809, 812
 \glssymbolnav ... 273, 595, 848, 1032
 \GLSsymbolplural . 150, 458, 710, 964
 \Glssymbolplural
 150, 150, 458, 710, 963
 \glssymbolplural
 149, 150, 458, 710, 963
 \glssymbolpluralaccessdisplay .
 345, 358, 544,
 546, 547, 556, 797, 799, 800, 810
 \glssymbolsgroupname . 40, 44, 47,
 235, 393, 396, 397, 506, 644,
 646, 648, 758, 899, 902, 903, 1013
 \glstarget 104, 231,
 275–282, 284, 292, 293, 295,
 303, 305, 307, 310–312, 314,
 318, 319, 321, 323, 324, 504–
 506, 579–588, 596–600, 604–
 606, 611–613, 615–617, 619–
 623, 756–758, 832–841, 849–
 854, 858–860, 865–867, 869–
 871, 873–876, 937, 938, 1011
 \glstexorpdfstring 165,
 165, 166, 170, 204, 971, 974, 996
 \GLStext 142, 455, 707, 960
 \Glstext 142, 142, 455, 707, 960
 \glstext 142, 142, 455, 707, 960
 \glstextaccessdisplay . 345, 349,
 350, 355, 544, 547, 548, 550,
 551, 555, 797, 801, 803, 804, 808
 \glstextformat
 114, 122, 124, 435, 443,
 444, 686, 694, 695, 943, 947, 948
 \glstextup 45, 241, 369, 396,
 509, 566, 646, 761, 819, 902, 1016
 \glstildechar 56, 178,
 182, 183, 402, 471, 474, 652,
 723, 725, 726, 908, 977, 980, 981
 \glstranslatefalse
 .. 30, 31, 388, 389, 638, 892, 893
 \glstranslatetrue 30,
 31, 388, 389, 638, 639, 892, 893
 \glstreechildpredesc
 317, 318, 319, 619, 620, 872–874
 \glstreegroupheaderfmt
 297–302, 316,
 318–322, 325, 608–611, 618–
 621, 623, 861–864, 872–875, 877
 \glstreeindent
 319, 320, 321, 323, 324, 583–
 585, 620–623, 837, 838, 874, 876
 \glstreeitem .. 297, 298, 317, 317,
 607, 608, 618, 619, 861, 862, 872

<code>\glstreenamebox</code>	521, 560, 561, 564–566, 764–
322, 323, 324, 622, 623, 875, 876	770, 772, 813–815, 817, 819,
<code>\glstreenamefmt</code> .. 316, 316, 318,	820, 1019, 1021–1023, 1025–1027
319, 321–324, 618–623, 872–876	<code>\GLSuseri</code>
<code>\glstreenavigationfmt</code>	151, 459, 710, 964
..... 298–302, 316, 319–	<code>\Glsuseri</code> 151, 151, 459, 710, 964
321, 325, 608–610, 618, 620,	<code>\glsuseri</code> 150, 151, 458, 710, 964
621, 623, 861–864, 872–875, 877	<code>\glsuseriaccessdisplay</code>
<code>\glstreepredesc</code> 346, 358, 545, 557, 798, 810
317,	<code>\GLSuserii</code>
318, 319, 321, 619–621, 872–874	152, 459, 711, 964
<code>\glstreesubitem</code>	<code>\Glsuserii</code> ... 152, 152, 459, 711, 964
297, 317, 317, 607, 619, 861, 872	<code>\glsuserii</code> ... 151, 152, 459, 710, 964
<code>\glstreesubsubitem</code>	<code>\glsuseriiaccessdisplay</code>
297, 317, 317, 607, 619, 861, 872 346, 359, 545, 557, 798, 810
<code>\glstype</code>	<code>\GLSuseriii</code>
122, 123, 136–	153, 459, 711, 965
141, 157–164, 353–355, 443,	<code>\Glsuseriii</code> ... 153, 153, 459, 711, 965
452–455, 461–465, 553, 554,	<code>\glsuseriii</code> 152, 153, 459, 711, 964, 965
694, 704–706, 712–717, 806–	<code>\glsuseriiiaccessdisplay</code>
808, 947, 948, 958–960, 966–971 347, 359, 545, 557, 798, 810
<code>\glsucmarkfalse</code> ... 14, 378, 628, 882	<code>\GLSuseriv</code>
<code>\glsucmarktrue</code> 14, 378, 628, 882	154, 460, 711, 965
<code>\glsunexpandedfieldvalue</code>	<code>\Glsuseriv</code> ... 154, 154, 460, 711, 965
..... 92, 277, 674, 850, 930	<code>\glsuseriv</code> ... 153, 154, 459, 711, 965
<code>\glsunset</code> 105, 107, 109, 110,	<code>\glsuserivaccessdisplay</code>
120, 430–433, 452–455, 681–	347, 359, 545, 557, 798, 810, 811
684, 704–707, 938, 939, 941, 946	<code>\GLSuseriv</code>
<code>\glsunsetall</code>	155, 460, 712, 966
107, 432, 682, 939	<code>\Glsuseriv</code> 155, 155, 460, 712, 965
<code>\glsupacpluralsuffix</code>	<code>\glsuseriv</code> 154, 155, 460, 711, 965
..... 45, 250, 251, 257,	<code>\glsuserivaccessdisplay</code>
396, 515, 520, 524, 527, 529,	347, 360, 545, 557, 558, 799, 811
532, 646, 767, 772, 777, 780,	<code>\GLSuserivi</code>
782, 785, 902, 1021, 1022, 1027	156, 460, 712, 966
<code>\glsupercase</code> 4, 52, 90, 115–	<code>\Glsuserivi</code> ... 156, 156, 460, 712, 966
118, 142–156, 158, 160, 162,	<code>\glsuserivi</code> ... 155, 156, 460, 712, 966
164, 246, 253–256, 333, 334,	<code>\glsuseriviaccessdisplay</code>
349–352, 354–360, 365–367, 347, 360, 545, 558, 799, 811
369, 877, 905, 928, 943–945,	<code>\GlsWarnAddProtectedPageFmt</code> ...
960–969, 971, 1018, 1024–1026 203, 204, 995
<code>\GlsUseAcrEntryDispStyle</code>	<code>\glswrallowprimitivemodsfalse</code> .
..... 247, 248, 205, 996
250–253, 255, 257, 258, 363–	<code>\glswrallowprimitivemodstrue</code> ..
365, 367, 369, 370, 512–518, 489, 741
520, 560, 561, 564–566, 764–	<code>\glswrglossdisableanchorcmds</code> ..
768, 770, 772, 813–815, 817,	... 204, 208, 238, 996, 997, 1015
819, 1019, 1021–1023, 1025–1027	<code>\glswrglossdisablelocationcmds</code>
<code>\GlsUseAcrStyleDefs</code> 199, 202, 203, 993, 995
..... 247, 248, 250–253, 255,	<code>\glswrglosslocationtarget</code>
257, 258, 363–365, 367, 369– 204, 238, 996, 1015
371, 512, 513, 515–518, 520,	<code>\glswrglosslocationtextfmt</code>
 204, 238, 996, 1015
	<code>\glswrite</code> 180–185, 193, 195,
	200, 472–476, 482, 486, 487,

	574–578, 724–728, 734, 738, 739, 828–831, 978–982, 989, 994		
<code>\glswritedefhook</code>	86, 94, 419, 424, 669, 675, 925, 931		<code>\if@endfor</code> 594, 847, 1031
<code>\glswriteentry</code>	202, 202, 487, 739, 994, 995		<code>\if@filesw</code> 34, 895
<code>\glswritefiles</code>	38, 200, 392, 486, 642, 738, 897, 898, 993		<code>\if@gl@debug</code> 7, 23, 201, 375, 383, 487, 624, 633, 739, 878, 887, 994
<code>\glxindyfalse</code>	33, 390, 640, 894		<code>\if@gl@docloaded</code> . 6, 6, 19, 374, 381, 393, 624, 631, 643, 878, 885
<code>\glxindytrue</code>	33, 34, 390, 640, 894, 895		<code>\if@gl@isacronymlist</code> 22, 122, 383, 443, 632, 694, 887, 947
H			<code>\if@openright</code> 53, 400, 650, 906
<code>\H</code>	26, 386, 635, 890		<code>\if@org@KV@gl@link@local</code> . . 120, 946
<code>\hangindent</code>	301, 302, 317, 319, 321, 323–325, 583– 585, 610, 611, 618, 620–623, 836–838, 864, 872–874, 876, 877		<code>\ifbool</code> 38, 65, 382, 389, 392, 408, 435–437, 631, 639, 642, 658, 686–688, 898, 913
<code>\hbox</code>	105, 279, 430, 580, 598, 681, 833, 851, 938		<code>\ifboolexpr</code> 46, 71, 396, 411, 506, 646, 661, 758, 902, 917
<code>\hfill</code>	279, 580, 598, 833, 851		<code>\ifcase</code> 7, 9, 10, 30, 36, 39, 41, 78, 226, 318, 375– 377, 388, 393, 414, 502, 583, 619, 624, 626, 627, 638, 643, 664, 753, 836, 873, 878, 880, 881, 892, 896, 898, 900, 920, 1009
<code>\hline</code>	281–283, 285, 292–294, 296, 304–306, 308, 309, 311– 316, 599–601, 605–607, 611– 618, 852–854, 858–861, 865–872		<code>\ifcsdef</code> 30, 47, 53, 81, 82, 89–94, 104, 202, 217, 220– 222, 247, 248, 343, 388, 397, 400, 416, 417, 421–424, 441, 487, 495, 497, 498, 507, 512, 543, 638, 647, 650, 666, 667, 671–675, 692, 739, 747, 749, 750, 759, 764, 796, 892, 903, 906, 922, 923, 927–931, 937, 994, 1002, 1004–1006, 1019, 1031
<code>\hsize</code>	279, 280, 291, 303, 310, 598, 604, 611, 615, 851, 852, 858, 864, 865, 868, 869		<code>\ifcsequal</code> 67, 329, 330, 409, 590, 659, 843, 844, 915
<code>\hspace</code>	317, 619, 872		<code>\ifcsequal</code> 68, 409, 659, 915
<code>\hss</code>	279, 580, 598, 833, 851		<code>\ifcsname</code> 93, 674, 930
<code>\ht</code>	290, 603, 857		<code>\ifcsstrequal</code> 94, 424, 675, 931
<code>\hyperbf</code>	239, 509, 761, 1015		<code>\ifcsstring</code> 93, 423, 674, 930
<code>\hyperdef</code>	43, 395, 645, 901		<code>\ifcsundef</code> . 10, 17, 33, 43, 46, 49, 51, 53, 64, 65, 73, 74, 77, 96, 98, 99, 108, 119, 125, 134, 135, 200, 212, 218, 226, 230, 234– 237, 240, 247, 344–347, 377, 380, 390, 395, 396, 398–400, 407, 408, 411–413, 425, 427, 432, 442, 444, 451, 452, 480, 481, 486, 492, 494, 495, 501, 504, 506, 507, 509, 512, 544– 546, 578, 594, 626, 630, 640, 645, 647, 649, 650, 658, 662, 664, 676, 678, 683, 693, 695,
<code>\hyperemph</code>	239, 509, 761, 1015		
<code>hyperfirst</code> (option)	31		
<code>\hyperit</code>	239, 509, 761, 1015		
<code>\hyperlink</code>	120, 134, 238, 442, 451, 508, 693, 703, 760, 946, 957, 1015		
<code>\hypermd</code>	239, 509, 761, 1015		
<code>\hypermm</code>	239, 509, 760, 1015		
<code>\hypersc</code>	239, 509, 761, 1015		
<code>\hypersf</code>	239, 509, 761, 1015		
<code>\hypersl</code>	239, 509, 761, 1015		
<code>\hypertarget</code>	134, 451, 703, 957		
<code>\hypertt</code>	239, 509, 761, 1015		
<code>\hyperup</code>	239, 509, 761, 1015		
I			
<code>\IeC</code>	26, 385, 635, 889		
<code>\if</code>	126, 208, 445, 490, 574, 696, 742, 827, 949, 950, 997		

703, 732, 738, 744, 746, 747, 753, 756, 758, 759, 761, 764, 797–799, 832, 848, 881, 884, 894, 901, 902, 904–906, 913, 918, 919, 932, 934, 940, 946, 948, 949, 957, 993, 999, 1003, 1008, 1009, 1011–1015, 1019, 1031	500, 634, 635, 679, 751, 752, 889, 894, 902, 935, 986, 987, 1007
<code>\ifcvoid</code> 69, 915	<code>\ifdim</code> 249, 290, 322, 514, 603, 622, 766, 857, 875, 1021
<code>\ifdef</code> 5, 42, 69, 79, 85, 86, 105, 120, 165, 166, 197, 198, 204, 241, 276, 317, 330, 343, 410, 415, 418, 430, 442, 465, 467, 484, 485, 509, 543, 591, 618, 619, 660, 665, 669, 681, 693, 717, 719, 736, 737, 761, 796, 844, 850, 872, 877, 900, 916, 921, 925, 938, 946, 971, 991, 992, 996, 1016	<code>\iffalse</code> 100, 106, 428, 431, 679, 682, 935, 939
<code>\ifdefempty</code> . 22, 35, 52, 53, 64, 68– 70, 74, 114, 116, 192, 193, 195, 222–224, 245, 247, 253, 262, 348, 350, 365, 383, 391, 400, 407, 409, 410, 412, 435, 438, 439, 482, 483, 499, 500, 511, 512, 517, 522, 526, 528, 530, 546, 549, 551, 562, 633, 641, 650, 657, 659, 660, 662, 663, 686, 689, 690, 734, 735, 751, 752, 763, 764, 769, 775, 778, 780, 783, 799, 802, 804, 815, 887, 896, 906, 913, 915, 916, 918, 919, 943, 944, 988–990, 1006, 1007, 1018, 1019, 1023, 1028	<code>\IfFileExists</code> 12, 29, 30, 191–193, 215, 378, 388, 480, 481, 494, 628, 637, 638, 732, 733, 745, 882, 892, 987–989, 1001
<code>\ifdefequal</code> 34, 67–70, 81, 82, 87, 97, 101, 224, 344, 390, 391, 408–410, 416, 417, 419, 426, 429, 500, 544, 640, 641, 659, 660, 666, 667, 670, 677, 680, 752, 797, 895, 914– 916, 922, 923, 926, 933, 936, 1007	<code>\ifglossaryexists</code> 50, 62, 64, 66, 189–192, 399, 406– 408, 479–481, 649, 656–658, 731–733, 904, 912–914, 986–988
<code>\ifdefstrequal</code> 93, 423, 674, 930	<code>\ifgl@sanitize@description</code> 27, 386, 636, 890
<code>\ifdefstring</code> 13, 46, 71, 190–192, 195, 219–221, 223, 225, 335, 378, 396, 411, 480– 482, 497–501, 538, 628, 646, 661, 731–733, 735, 748, 749, 751, 752, 791, 882, 902, 917, 986–988, 990, 1004–1006, 1008	<code>\ifgl@sanitize@name</code> 27, 386, 636, 890
<code>\ifdefvoid</code> 25, 26, 33, 45, 100, 190, 191, 223, 224, 385, 428, 499,	<code>\ifgl@sanitize@symbol</code> 28, 386, 636, 891
	<code>\ifgl@s@xindy@glnumbers</code> 63, 407, 657, 912
	<code>\ifgl@sacrdescription</code> 32, 533, 787, 894
	<code>\ifgl@sacrdua</code> 32, 524, 529, 532–534, 777, 782, 785, 787, 894
	<code>\ifgl@sacrfootnote</code> 31, 122, 443, 533, 534, 694, 787, 893, 947
	<code>\ifgl@sacracronym</code> . 20, 381, 382, 631, 886
	<code>\ifgl@sacrshortcuts</code> 42, 259, 394, 521, 645, 773, 900, 1028
	<code>\ifgl@sacrsmalldcaps</code> 32, 524, 525, 527, 529, 532, 776–778, 780, 782, 784, 785, 894
	<code>\ifgl@sacrsmaller</code> . . 32, 524, 525, 528, 529, 776–778, 780, 782, 894
	<code>\ifgl@sautomake</code> 36, 37, 42, 195, 391, 392, 483, 641, 642, 734, 896, 897, 900, 990
	<code>\ifgl@sdescsuppressed</code> 68, 274, 409, 595, 659, 849, 915
	<code>\ifgl@sentrycounter</code> 14, 15, 42, 229, 230, 379, 395, 503, 504, 628, 629, 645, 755, 756, 882, 883, 900, 1010, 1011
	<code>\ifgl@sentryexists</code> . . . 65, 65, 66, 83–85, 95, 97, 407, 408, 418, 424, 426, 658, 668, 669, 675, 677, 913, 914, 924, 925, 931, 933
	<code>\ifgl@sesclocations</code> 127, 202–204, 488, 740, 950, 995, 996
	<code>\ifgl@sfieldcseq</code> . . . 94, 424, 675, 931

<code>\ifglsgfielddefeq</code> .. 93, 423, 674, 930	<code>\ifglssanitizesort</code> 25, 28,
<code>\ifglsgfieldefeq</code> 93, 423, 674, 930	384, 387, 634, 636, 637, 888, 891
<code>\ifglsgfieldvoid</code> 68, 915	<code>\ifglssavenumberlist</code>
<code>\ifglsgshaschildren</code> 67, 275,	81, 194, 211, 416, 483,
408, 579, 596, 658, 832, 849, 914	492, 666, 734, 743, 922, 989, 999
<code>\ifglsgshasdesc</code>	<code>\ifglssavewrites</code>
. 67, 274, 409, 596, 659, 849, 915	38, 189, 202, 392, 479,
<code>\ifglsgshasfield</code> 69, 410, 660, 915	487, 642, 731, 739, 897, 986, 994
<code>\ifglsgshaslong</code> .. 68, 111–113, 166,	<code>\ifglsgsubentrycounter</code> 15, 42, 229,
167, 248, 249, 253, 255, 362,	230, 379, 395, 503, 504, 629,
365, 368, 409, 434, 435, 466,	645, 755, 756, 883, 900, 1010, 1011
513, 514, 517, 519, 528, 559,	<code>\ifglsgstoc</code> 54, 401, 651, 906
560, 562, 564, 659, 685, 686,	<code>\ifglsgstranslate</code> 29,
717, 765, 766, 769, 770, 781,	45, 388, 396, 637, 646, 892, 902
812, 813, 815, 817, 915, 942,	<code>\ifglsgsucmark</code> 51, 52, 399, 649, 650, 905
943, 972, 1019–1021, 1024, 1025	<code>\ifglsgsused</code> 65, 110, 114–116,
<code>\ifglsgshasparent</code> 67, 218,	122, 177, 202, 330–334, 348–
223, 322, 409, 495, 499, 621,	350, 408, 434–439, 443, 471,
659, 747, 751, 875, 915, 1003, 1007	487, 523, 526, 528, 530, 546–
<code>\ifglsgshasprefix</code> 329,	551, 591–593, 658, 684, 686–
330, 332, 333, 590–593, 843–846	690, 694, 722, 739, 775, 778,
<code>\ifglsgshasprefixfirst</code> 329,	781, 783, 799–804, 844–847,
330, 332, 333, 590–593, 844–846	913, 941, 943, 944, 947, 977, 995
<code>\ifglsgshasprefixfirstplural</code>	<code>\ifglsgswrallowprimitivemods</code>
..... 330, 331, 333, 205, 206, 489, 741, 996
334, 590, 591, 593, 594, 844–847	<code>\ifglsgsxindy</code>
<code>\ifglsgshasprefixplural</code> . 329, 331,	32, 48, 49, 54–63, 74, 101, 102,
332, 334, 590–593, 843, 845–847	127, 178, 179, 185, 190, 191,
<code>\ifglsgshasshort</code>	193, 208, 209, 215, 390, 398,
.... 68, 211, 409, 660, 915, 999	401–407, 412, 429, 446, 471,
<code>\ifglsgshassymbol</code> 68,	472, 476, 479, 480, 490, 491,
318, 319, 321, 323, 324, 409,	494, 573, 574, 640, 648, 651–
523, 526, 530, 619–623, 659,	657, 662, 680, 697, 723, 724,
775, 778, 783, 873, 874, 876, 915	728, 731, 732, 741, 743, 746,
<code>\ifglsgshyperfirst</code> . 122, 443, 694, 947	826–828, 883, 894, 904, 907–
<code>\ifglsgsindexonlyfirst</code>	912, 918, 936, 950, 977, 978,
..... 202, 487, 739, 995	983, 986, 987, 989, 997, 998, 1001
<code>\ifglsgsnogroupskip</code> 277,	<code>\ifglsgsignoredglossary</code> 75,
281, 282, 284, 287, 288, 292,	96, 100, 202, 212, 413, 425,
294, 295, 303, 305, 307, 311,	428, 487, 492, 663, 676, 679,
313, 315, 318, 320, 321, 324,	739, 744, 919, 932, 935, 994, 1000
597, 599, 600, 602, 605, 606,	<code>\ifglsgs@</code> 41, 394, 644, 900
611–613, 615–617, 619–621,	<code>\ifglsgs@listcs</code> 221,
623, 850, 852–856, 858–860,	226, 498, 501, 750, 753, 1005, 1008
865–867, 869–871, 873–875, 877	<code>\ifglsgs@finer</code> 8, 375, 625, 879
<code>\ifglsgsnonumberlist</code> 228, 502, 754, 1010	<code>\ifglsgs@KV@glsgslink@hyper</code>
<code>\ifglsgsnoopostdot</code> ... 13, 378, 628, 882	123, 124, 443, 444, 694, 695, 948
<code>\ifglsgsnumberline</code> .. 54, 401, 651, 906	<code>\ifglsgs@KV@glsgslink@local</code>
<code>\ifglsgsresetcurrcount</code> .. 108, 109, 940 120, 452–455, 704–706, 946
	<code>\ifglsgs@measuring@</code> 103, 430, 681, 937

<code>\ifmmode</code>	8, 375, 625, 879	956, 972, 978, 981–983, 985, 993, 995–1001, 1013–1015, 1028
<code>\ifnum</code>	16, 109, 110, 192, 223, 290, 319, 321, 323, 324, 380, 391, 433, 482, 499, 583–585, 603, 620–622, 629, 630, 642, 683, 684, 733, 751, 837, 838, 857, 874, 876, 884, 940, 941, 988, 1006	<code>\immediate</code> 35, 84, 86, 110, 190– 193, 200, 216, 391, 418, 419, 434, 479, 481, 487, 494, 641, 668, 669, 684, 685, 731, 733, 738, 739, 746, 896, 924, 925, 941, 942, 986–989, 994, 1001, 1002
<code>\ifstrempty</code>	9, 376, 579, 626, 832, 880	<code>\in@</code> 41, 394, 644, 900
<code>\ifstrequal</code>	506, 758	<code>index (option)</code> 40
<code>\ifthenelse</code>	29, 43, 53, 125, 194, 387, 395, 400, 444, 482, 506, 534, 594, 637, 645, 650, 695, 734, 758, 787, 847, 891, 901, 906, 948, 989, 1031	<code>index (style)</code> 317
<code>\IfTrackedLanguage</code>	186, 476, 728, 983	<code>indexgroup (style)</code> 318
<code>\IfTrackedLanguageFileExists</code>	46, 326, 327, 397, 572, 589, 647, 825, 842, 903	<code>indexhypergroup (style)</code> 318
<code>\iftrue</code>	100, 106, 428, 431, 679, 682, 935, 939	<code>\indexname</code> 41, 394, 644, 899
<code>\ifundef</code>	14, 15, 72, 84, 96, 180, 184, 193, 379, 411, 418, 425, 472, 475, 482, 628, 629, 661, 668, 676, 724, 727, 734, 882, 883, 917, 924, 932, 978, 981, 989	<code>indexonlyfirst (option)</code> 31
<code>\ifvmode</code>	177, 470, 722, 977	<code>\indexspace</code> 276 , 277 , 297 , 297 , 298 – 302, 316 , 318–322, 324, 325, 596, 597, 607–611, 618–621, 623, 850, 861–864, 872–875, 877
<code>\ifvoid</code>	290, 604, 857	<code>inline (style)</code> 273
<code>\ifx</code>	14, 16, 18, 21, 35, 40, 41, 43, 55, 56, 58, 60, 62, 63, 71, 97–99, 102, 104, 125, 128–134, 167, 180, 183, 185, 186, 188, 200, 204, 207, 209, 210, 213, 214, 235, 237, 238, 262, 263, 335, 344, 379–383, 394, 395, 401, 402, 404, 406, 407, 425–429, 444, 446–451, 466, 472, 474, 476–478, 486, 488–491, 493, 506–508, 522, 524, 525, 527, 529–531, 533, 535, 537, 543, 574–579, 583– 585, 628–630, 632, 641, 644, 645, 651, 652, 654, 655, 657, 676–680, 695, 697–703, 717, 718, 724, 726, 728–730, 738, 740–743, 745, 758–760, 774, 776, 778, 780, 782–784, 786, 788, 791, 797, 828, 830–832, 836–838, 882–887, 896, 899– 901, 907, 908, 910, 912, 913, 917, 932–937, 948, 949, 951–	<code>\input</code> 45, 113, 396, 435, 646, 686, 902, 943
		<code>\inputencodingname</code> 33, 390, 640, 894
		<code>\InputIfFileExists</code> 85, 418, 669, 925
		<code>\istfile</code> 199 , 486, 738
		<code>\istfilename</code> 48, 48 , 180, 184, 191, 192, 194, 398, 472, 475, 480–482, 574, 577, 648, 724, 727, 732–734, 828, 830, 904, 978, 981, 987–989
		<code>\item</code> 276–280, 297, 298, 317–319, 579, 580, 582, 583, 597, 598, 607, 608, 619, 620, 832, 833, 836, 850–852, 861, 862, 872, 873
J		
		<code>\jobname</code> 35, 48, 84, 85, 180, 184, 190–193, 215, 391, 398, 418, 472, 475, 479–481, 494, 574, 577, 641, 648, 668, 669, 724, 727, 731– 733, 745, 828, 830, 896, 904, 924, 925, 978, 981, 986–989, 1001
K		
		<code>kernelglossredefs (option)</code> 39
		<code>\key@ifundefined</code> 87–89, 420, 670, 671, 926, 927
		keys: <code>access</code> 336

counter	77	
description	75	
descriptionaccess	337	
descriptionplural	75	
descriptionpluralaccess	337	
first	76	
firstaccess	336	
firstplural	76	
firstpluralaccess	336	
long	80	
longaccess	337	
longplural	80	
longpluralaccess	337	
name	75	
nonumberlist	78	
parent	78	
plural	76	
pluralaccess	336	
see	77	
short	80	
shortaccess	337	
shortplural	80	
shortpluralaccess	337	
sort	76	
symbol	76	
symbolaccess	336	
symbolplural	77	
symbolpluralaccess	336	
text	76	
textaccess	336	
type	77	
user1	79	
user1access	337	
user2	79	
user2access	337	
user3	79	
user3access	337	
user4	79	
user4access	338	
user5	79	
user5access	338	
user6	80	
user6access	338	
\KV@glslink@hyperfalse	120, 122, 123, 135, 442, 443, 452, 693, 694, 703, 946–948, 957	
\KV@glslink@hypertrue	120, 135, 442, 452, 693, 703, 946, 957	
		L
\L	27, 386, 635, 890	
\l	5, 27, 386, 635, 878, 890	
\label	11, 103, 104, 227, 229, 377, 502, 503, 627, 754, 755, 881, 937, 938, 1009, 1010	
\languagename	33, 390, 640, 894	
languages (option)	31	
\leaders	279, 580, 598, 833, 851	
\leavevmode	94, 123, 424, 443, 675, 694, 931, 948	
\let	6, 7, 13, 16–18, 20, 26, 27, 29–31, 33, 35–43, 46, 47, 57, 69–71, 81, 83, 94, 95, 97, 99, 100, 103–106, 108, 110, 113, 120–124, 126, 127, 134–141, 157–164, 167, 175, 184, 185, 189–192, 194, 196–198, 201, 203, 204, 206, 210–215, 224, 226, 228, 245, 258, 259, 272, 273, 276, 277, 290, 297, 298, 317, 326, 328, 338, 339, 353–355, 374, 375, 378, 380–382, 385, 386, 388, 389, 391–396, 398, 402, 403, 410, 411, 416, 417, 424–428, 430–433, 435, 442–446, 451–455, 461–466, 469, 470, 475, 476, 479–485, 487–489, 491–493, 500, 501, 503, 505, 511, 521–533, 540, 553, 554, 567–572, 578, 589, 590, 594, 595, 603, 607, 608, 619, 624, 625, 628–631, 635–639, 641–648, 653, 660, 661, 666–668, 675–679, 681–684, 686, 693–697, 703–707, 712–718, 721, 722, 726, 728, 731–737, 739–741, 743–745, 752, 753, 755, 757, 763, 773, 774, 776–784, 786, 793, 794, 806–808, 820–825, 831, 842, 843, 847, 848, 850, 857, 861, 862, 872, 878, 879, 882–886, 889, 890, 892–903, 908, 909, 916, 917, 922–924, 931–935, 937–941, 943, 946–948, 950, 957–960, 966–972, 976, 981, 982, 986–996, 999–1001, 1007–1010, 1018, 1027, 1028, 1031, 1032	
\letcs	33, 35, 67–70, 86, 88,	

89, 92, 98, 99, 165, 167, 197, 198, 217–219, 223, 224, 231, 234, 322, 408–410, 418, 420, 423, 427, 465, 466, 480, 481, 484, 485, 495, 496, 499, 500, 504, 506, 622, 641, 659, 660, 669, 671, 674, 677, 678, 717, 732, 733, 736, 737, 747, 748, 751, 752, 756, 758, 875, 894, 896, 914–916, 925, 927, 930, 933, 934, 971, 972, 991, 992, 1002–1004, 1007, 1011–1013, 1031	
<code>list</code> (style)	277
<code>\listcsadd</code>	221, 498, 750, 1005
<code>\listcsgadd</code>	226, 501, 753, 1008
<code>\listcsxadd</code>	217, 495, 747, 1002
<code>listdotted</code> (style)	279
<code>\listead</code>	221, 498, 750, 1005
<code>listgroup</code> (style)	277
<code>listhypergroup</code> (style)	278
<code>\listxadd</code>	84, 668, 924
<code>\loadglsentries</code> 113, 113, 435, 686, 943	
<code>\long</code>	94, 238, 424, 509, 675, 760, 931, 1015
<code>long</code> (key)	80
<code>long</code> (style)	280
<code>long-booktabs</code> (style)	287
<code>long-sc-short</code> (acrstyle)	250
<code>long-sc-short-desc</code> (acrstyle) 251, 363	
<code>long-short</code> (acrstyle)	248, 362
<code>long-short-desc</code> (acrstyle)	251, 363
<code>long-sm-short</code> (acrstyle)	250
<code>long-sm-short-desc</code> (acrstyle) 252, 364	
<code>long-sp-short</code> (acrstyle)	249
<code>long-sp-short-desc</code> (acrstyle)	251
<code>long3col</code> (style)	282
<code>long3col-booktabs</code> (style)	287
<code>long3colborder</code> (style)	283
<code>long3colheader</code> (style)	283
<code>long3colheaderborder</code> (style)	283
<code>long4col</code> (style)	283
<code>long4col-booktabs</code> (style)	288
<code>long4colborder</code> (style)	284
<code>long4colheader</code> (style)	284
<code>long4colheaderborder</code> (style)	285
<code>longaccess</code> (key)	337
<code>longborder</code> (style)	281
<code>longheader</code> (style)	281
<code>longheaderborder</code> (style)	282
<code>\longnewglossaryentry</code>	
.	94, 95, 424, 675, 931
<code>longplural</code> (key)	80
<code>longpluralaccess</code> (key)	337
<code>\longprovideglossaryentry</code>	
.	95, 424, 675, 931
<code>longragged</code> (style)	291
<code>longragged-booktabs</code> (style)	289
<code>longragged3col</code> (style)	293
<code>longragged3col-booktabs</code> (style)	289
<code>longragged3colborder</code> (style)	294
<code>longragged3colheader</code> (style)	294
<code>longragged3colheaderborder</code> (style)	294
<code>longraggedborder</code> (style)	292
<code>longraggedheader</code> (style)	292
<code>longraggedheaderborder</code> (style)	293
<code>\LT@end@pen</code>	290, 603, 857
<code>\LT@err</code>	290, 603, 857
<code>\LT@foot</code>	290, 603, 604, 857
<code>\LT@head</code>	290, 291, 603, 604, 857
<code>\LT@lastfoot</code>	290, 603, 604, 857
<code>\LT@output</code>	290, 603, 857
M	
<code>\makeatletter</code>	85, 215, 418, 494, 669, 745, 925, 1001
<code>\makebox</code>	279, 322–324, 580, 584, 585, 598, 622, 623, 833, 838, 851, 875, 876
<code>\makefirsttuc</code>	4, 877
<code>\makeglossaries</code>	9, 34–36, 39, 40, 44, 78, 189, 193, 195, 197, 199, 216, 376, 390–393, 396, 414, 479, 482– 484, 486, 494, 626, 640–643, 646, 664, 731, 734–736, 738, 746, 880, 895, 896, 898, 899, 901, 920, 986, 989–991, 993, 1002
<code>\makeglossary</code>	38, 38, 39, 40, 392, 393, 642, 643, 898, 899
<code>makeindex</code> (option)	33
<code>\makenoidxglossaries</code>	9, 34, 35, 78, 194, 195, 199, 376, 390, 391, 414, 483–486, 626, 640, 641, 664, 734–738, 880, 895, 920, 990–993
<code>\MakeTextUppercase</code>	5, 374, 624, 878
<code>\MakeUppercase</code>	547, 548, 554, 800–802, 807, 808

<code>\marginpar</code>	8, 376, 625, 879	<code>\NeedsTeXFormat</code>	4, 269, 273, 276, 280,
<code>\markboth</code>	51, 52, 399, 649, 905		286, 291, 297, 302, 309, 316,
<code>\mbox</code>	177, 278, 301, 302,		325–327, 334, 373, 374, 537,
	323, 470, 580, 597, 610, 622,		572, 573, 578, 589, 624, 790,
	722, 833, 851, 863, 864, 876, 977		825, 826, 832, 842, 850, 877, 1031
<code>mcolalmtree (style)</code>	301	<code>\new@glossaryentry</code>	83, 84, 197,
<code>mcolalmtreegroup (style)</code>	301		417, 418, 484, 668, 736, 924, 991
<code>mcolalmtreehypergroup (style)</code>	301	<code>\new@ifnextchar</code>	72,
<code>mcolalmtreespannav (style)</code>	302		89, 90, 111, 112, 136–140, 142–
<code>mcolindex (style)</code>	297		164, 243, 244, 330–334, 411,
<code>mcolindexgroup (style)</code>	297		421, 422, 434, 452–465, 509,
<code>mcolindexhypergroup (style)</code>	298		510, 591–593, 661, 672, 685,
<code>mcolindexspannav (style)</code>	298		704–716, 761, 762, 844–846,
<code>mcoltree (style)</code>	298		917, 928, 942, 957–970, 1016, 1017
<code>mcoltreegroup (style)</code>	299	<code>\newacronym</code>	240, 240, 241,
<code>mcoltreehypergroup (style)</code>	299		245, 509, 511, 522, 524, 525,
<code>mcoltreenoname (style)</code>	300		527, 529, 531, 533, 535, 761,
<code>mcoltreenonamegroup (style)</code>	300		763, 774, 776, 778, 780, 782,
<code>mcoltreenonamehypergroup (style)</code>	300		784, 786, 788, 1015, 1016, 1018
<code>mcoltreenonamespannav (style)</code>	300	<code>\newacronymhook</code>	245, 246, 371, 511, 522,
<code>mcoltreespannav (style)</code>	299		524, 526, 527, 529, 532, 533,
<code>\memUchead</code>	51, 399, 649, 905		535, 567, 763, 774, 776, 778,
<code>\MessageBreak</code>	23, 39, 71, 213,		780, 782, 785, 786, 788, 820, 1018
	214, 326, 327, 335, 384, 392,	<code>\newacronymstyle</code>	247, 248–253, 255,
	393, 411, 493, 537, 572, 589,		257, 258, 512–518, 520, 764–
	633, 642, 643, 661, 745, 791,		770, 772, 1019–1023, 1025–1027
	825, 826, 842, 888, 898, 917, 1000	<code>\newcommand</code>	4, 5, 8–33, 35–37, 39–45,
<code>mfirstuc (option)</code>	41		47–90, 93–95, 100–108, 110–
<code>\mfirstucMakeUppercase</code>	5, 374, 399, 422,		114, 116, 118–124, 126, 127,
	436–440, 455–465, 512, 517–		134–179, 185, 188–193, 195,
	519, 526, 530, 550–552, 555–		199–204, 206, 208–213, 215–
	558, 562, 563, 565, 593, 594,		225, 227–236, 239–249, 258,
	624, 649, 650, 672, 687–691,		260, 262–273, 275–277, 290,
	707–715, 717, 763, 764, 769–		297, 316, 317, 322, 328–334,
	771, 779, 783, 803–805, 808–		336, 340–344, 361, 371–373,
	811, 815, 816, 818, 846, 847, 878		375–385, 387–409, 411–424,
<code>\MFUaddmap</code>	5, 877		428–435, 438, 439, 441–446,
<code>\MFUblocker</code>	5, 877		451–472, 476, 479–483, 485–
<code>\MFUexcl</code>	5, 877		492, 494–507, 509–514, 521–
<code>\MFUsentencecase</code>	5, 5, 41, 165,		538, 542–544, 559, 567, 571,
	166, 170, 877, 878, 900, 971, 974		572, 578, 590–597, 603, 607,
<code>\midrule</code>	287, 288, 602, 855, 856		618, 619, 622, 625–635, 637–
<code>\month</code>	180, 184, 472, 475, 574,		639, 641–646, 648–653, 655–
	577, 724, 727, 828, 830, 978, 981		659, 661–675, 679–686, 689–
			697, 703–723, 728, 731–735,
			737–766, 773–791, 795–797,
			812, 820, 824, 825, 832, 843–
			850, 857, 861, 872, 875, 877–
N			
<code>\n</code>	184, 475,		
	577, 578, 727, 830, 831, 981, 982		
<code>name (key)</code>	75		

928, 930, 931, 935–944, 946–
 948, 950, 957–978, 983, 985–
 990, 993–1019, 1021, 1027–1032
 \newcount 17,
 81, 380, 416, 630, 666, 884, 922
 \newcounter
 . . 14, 15, 379, 628, 629, 882, 883
 \newenvironment . 230, 504, 756, 1011
 \newglossary
 . . 19, 20, 40, 41, 72, 73, 194,
 381, 382, 393, 394, 411, 412,
 482, 631, 632, 644, 661, 662,
 734, 885, 886, 899, 917, 918, 989
 \newglossaryentry 9,
 41, 80, 83, 83, 108, 246, 376,
 394, 415, 417, 432, 511, 521,
 523, 524, 526, 528, 530, 532,
 534, 567–570, 626, 644, 665–
 668, 683, 763, 773, 775, 777,
 779, 781, 783, 785, 787, 820–
 823, 880, 899, 921–924, 940, 1018
 \newglossarystyle 235, 273,
 277–289, 291–315, 317–322,
 325, 507, 595, 597–623, 759,
 848, 850–856, 858–875, 877, 1013
 \newif 6,
 7, 22, 29, 32, 36, 108, 205, 374,
 375, 383, 388, 390, 391, 489,
 624, 632, 637, 640, 641, 741,
 878, 887, 892, 894, 896, 940, 996
 \newignoredglossary 74, 412, 662, 918
 \newlength . . . 134, 279, 280, 291,
 303, 310, 320, 451, 598, 604,
 611, 615, 620, 703, 851, 852,
 858, 864, 865, 868, 869, 874, 957
 \newrobustcmd
 . . 8, 9, 67–69, 83, 84, 89–93,
 111, 112, 122, 136–166, 170,
 176, 177, 210, 211, 234, 237,
 241–244, 262, 322, 330–333,
 344–347, 376, 408–410, 417,
 418, 421, 422, 434, 442, 452–
 471, 491, 492, 509, 510, 544–
 546, 590–593, 621, 625, 626,
 658–660, 667, 668, 672, 685,
 693, 704–722, 743, 761, 762,
 797–799, 843–846, 875, 880,
 914, 915, 923, 924, 928–930,
 942, 947, 957–972, 974, 977,
 999, 1012, 1014, 1016, 1017, 1028
 \newterm 41, 394, 644, 899
 \newtoks 128, 189, 245, 446, 479, 511,
 697, 731, 763, 951, 986, 1017, 1018
 \newwrite . . 35, 84, 180, 184, 189,
 193, 391, 418, 472, 475, 479,
 482, 641, 668, 724, 727, 731,
 734, 896, 924, 978, 981, 986, 989
 \nfss@text 8, 375, 625, 879
 \noalign 290, 603, 857
 \nobreak 278,
 291, 580, 597, 604, 833, 851, 857
 \noexpand . . 21, 43, 55, 56, 85, 99,
 100, 104, 125, 127, 128, 133,
 134, 175, 186–188, 190–192,
 194, 203, 208, 211, 216, 218,
 219, 238, 240, 246, 382, 395,
 401, 402, 418, 427, 428, 441,
 442, 444–446, 451, 470, 476–
 481, 483, 488, 489, 492, 494,
 496, 504, 505, 509, 511, 521,
 523–528, 530–532, 534, 538,
 567–571, 573, 632, 646, 652,
 669, 678, 679, 692, 695–697,
 702, 703, 721, 728–734, 740,
 741, 744, 746, 748, 756, 757,
 761, 763, 773, 775–777, 779,
 781, 783–785, 787, 788, 791,
 820–824, 826, 886, 901, 907,
 908, 925, 934, 935, 937, 948–
 951, 956, 976, 983–990, 995,
 998, 999, 1001–1004, 1015, 1018
 noglossaryindex (option) 41
 nogroupskip (option) 14
 \nohyperpage . 237, 508, 759, 760, 1014
 nohypertypes (option) 23
 \noindent 298–
 302, 320–322, 505, 608–610,
 620, 621, 757, 862–864, 874, 875
 \noist 184,
 185, 189, 189, 479, 577, 578,
 731, 830, 831, 981, 982, 985, 986
 nolangwarn (option) 23
 nolist (option) 13
 nolong (option) 12
 nomain (option) 20
 nonumberlist (key) 78
 nonumberlist (option) 12
 \nopostdesc
 41, 47, 47, 94, 214, 394, 397,

	398, 424, 493, 579, 644, 648, 675, 745, 832, 899, 903, 931, 1001	
nopostdot (option)	13	
noredefwarn (option)	24	
\normalbaselineskip	290, 603, 857	
nostyles (option)	13	
nosuper (option)	13	
notranslate (option)	30	
notree (option)	13	
nowarn (option)	23	
\ns@ACRfull	243, 510, 762, 1017	
\ns@Acrfull	243, 510, 761, 1017	
\ns@acrfull	243, 509, 761, 1016	
\ns@ACRfullpl	244, 510, 762, 1017	
\ns@acrfullpl	244, 510, 762, 1017	
\ns@ACRlong	162, 464, 715, 969	
\ns@Acrlong	161, 463, 715, 969	
\ns@acrlong	160, 463, 714, 968	
\ns@ACRlongpl	164, 465, 716, 970	
\ns@Acrlongpl	163, 464, 716, 970	
\ns@acrlongpl	162, 464, 716, 970	
\ns@ACRshort	158, 461, 713, 967	
\ns@Acrshort	157, 461, 713, 966	
\ns@acrshort	156, 157, 461, 712, 966	
\ns@ACRshortpl	160, 462, 714, 968	
\ns@acrshortpl	159, 462, 714, 968	
\ns@newglossary	72, 72, 411, 661, 917	
\null	127– 134, 186–188, 215, 445–451, 476–479, 494, 696–703, 728– 730, 746, 950–956, 983–985, 1001	
\number	16, 85, 98, 104, 110, 203, 206, 207, 380, 418, 426, 433, 488, 489, 505, 538, 630, 669, 677, 684, 740, 741, 757, 791, 884, 925, 933, 937, 941, 995, 996	
numberedsection (option)	10	
\numberline	54, 401, 651, 907	
numberline (option)	10	
numbers (option)	40	
\numexpr	110, 433, 684, 941	
O		
\O	27, 386, 635, 890	
\o	27, 386, 636, 890	
\OE	27, 386, 635, 890	
\oe	27, 386, 635, 890	
\oldacronym	239, 509, 761, 1015	
\openout	35, 84, 180, 184, 190, 391, 418, 472, 475, 479, 574, 577, 641, 668, 724, 727, 731, 828, 830, 896, 924, 978, 981, 986	
options:		
acronym	20	
acronymlists	22	
acronyms	20	
automake	36	
automakegloss	37	
automakeglosslite	37	
compatible-2.07	38	
compatible-3.07	38	
counter	22	
counterwithin	14	
debug	7	
description	32	
disablemakegloss	34	
dua	32	
entrycounter	14	
esclocations	12	
footnote	31	
hyperfirst	31	
index	40	
indexonlyfirst	31	
kernelglossredefs	39	
languages	31	
makeindex	33	
mfirstuc	41	
noglossaryindex	41	
nogroupskip	14	
nohypertypes	23	
nolangwarn	23	
nolist	13	
nolong	12	
nomain	20	
nonumberlist	12	
nopostdot	13	
noredefwarn	24	
nostyles	13	
nosuper	13	
notranslate	30	
notree	13	
nowarn	23	
numberedsection	10	
numberline	10	
numbers	40	
order	32	
restoremakegloss	34	
sanitize	29	

sanitizesort	28	432, 442, 465, 479, 482, 484–
savenuumberlist	12	486, 497, 498, 501, 502, 506,
savewrites	38	507, 512, 513, 524, 525, 529,
section	10	532, 546, 579, 626, 631, 642,
seeautonumberlist	12	646, 652, 656, 658, 661, 664–
seenoindex	9	666, 671–677, 683, 693, 717,
shortcuts	32	731, 734, 736–738, 749, 750,
smallcaps	32	753, 754, 758, 759, 764, 765,
smaller	32	777, 778, 782, 785, 799, 832,
sort	15	880, 885, 890, 896, 901, 908,
style	11	912, 914, 917, 919, 921, 922,
subentrycounter	15	927–933, 940, 946, 971, 985,
symbols	40	986, 989, 991, 993, 1004–1006,
toc	10	1009, 1010, 1013, 1014, 1016, 1019
translate	30	\PackageInfo 7, 34, 35, 190, 201, 375,
ucmark	14	391, 479, 487, 624, 625, 640,
writeglslabelnames	36	641, 731, 739, 879, 895, 986, 994
writeglslabels	36	\PackageWarning 7, 23, 375,
xindy	33	383, 624, 625, 633, 878, 879, 887
xindygloss	34	\PackageWarningNoLine
xindynoglsnumbers	34 7, 23, 326, 327, 375,
\OR	534, 787	383, 384, 572, 589, 624, 625,
\or	7, 9–11, 30, 36,	633, 825, 842, 878, 879, 887, 888
	37, 39–41, 227, 318, 375, 377,	\pagegoal 290, 603, 857
	389, 393, 502, 583, 619, 624–	\pagelistname . . . 44, 47, 283–285,
	627, 638, 643, 753, 754, 836,	287, 288, 294, 296, 306–309,
	873, 879–881, 893, 897–900, 1009	313–316, 396, 397, 600–602,
order (option)	32	606, 607, 613–615, 617, 618,
\org@glossaryentrynumbers	214,	646, 647, 853, 854, 856, 859–
	228, 493, 503, 745, 755, 1001, 1010	861, 866–868, 870–872, 901, 903
\org@glossarytitle	213, 214, 493, 744, 745, 1000, 1001	\par 47, 276, 278,
\org@glspostdescription	47, 398, 648, 903	297, 299–302, 316, 317, 319–
\org@ifKV@glslink@hyper	123, 124, 443, 444, 694, 695, 948	325, 398, 505, 506, 580, 583–
\outputpenalty	290, 603, 857	585, 597, 607–611, 618–623,
		648, 757, 758, 833, 837, 838,
		850, 851, 861–864, 872–877, 903
		parent (key) 78
		\parindent 297–302,
		317, 319–321, 323–325, 583–
		585, 607–611, 619–623, 836–
		838, 861–864, 872–874, 876, 877
\p@	276, 297, 316, 317,	\parskip
	597, 607, 618, 619, 850, 861, 872	297–301, 317, 319, 320, 607–
\p@glshyp@opt	121, 121, 442, 693, 947	609, 619–621, 861–863, 872–874
\PackageError	9, 18, 19, 27, 36, 44,	\PassOptionsToPackage
	55, 62, 65, 66, 71, 77, 80, 81, 327, 334, 537, 589, 790, 842
	88–94, 96–98, 108, 119, 165,	\pdfstringdefPreHook 204, 996
	188, 189, 194, 197, 199, 219–	\penalty 290, 603, 857
	222, 226, 228, 235, 236, 241,	\PGLS 333, 593, 846
	247, 248, 347, 373, 374, 376,	\Pgls 331, 331, 592, 845
	381, 392, 396, 402, 406, 408,	
	411, 413, 415, 416, 420–426,	

<code>\pgls</code>	330, 331, 591, 844	195, 196, 202, 205, 207, 208,
<code>\PGLSpl</code>	333, 593, 846	211, 217, 218, 227, 229, 235,
<code>\PglSpl</code>	332, 332, 592, 845	238, 246, 262, 270, 343, 344,
<code>\pglSpl</code>	331, 332, 591, 844	371, 376, 377, 402, 403, 405,
<code>\phantomsection</code> ...	53, 400, 650, 906	407, 426, 428, 429, 435–437,
plural (key)	76	441, 444, 445, 470, 487–489,
pluralaccess (key)	336	502, 504, 505, 507, 511, 530,
<code>\printacronyms</code>	20, 382, 631, 886	534, 538, 543, 544, 574, 575,
<code>\PrintChanges</code>	6, 374, 624, 878	590, 594, 625, 627, 630, 632,
<code>\printglossaries</code>	195,	641, 652, 654, 656, 657, 659,
	213, 483, 492, 735, 744, 990, 1000	661–663, 673–677, 679, 680,
<code>\printglossary</code>	20, 24,	686–688, 692, 694–696, 721,
	40, 41, 195, 212, 213, 228, 382,	722, 735, 739–741, 744, 746,
	384, 393, 394, 483, 492, 502,	747, 754–757, 759, 763, 783,
	632, 634, 644, 735, 744, 754,	787, 791, 796, 797, 820, 827,
	886, 888, 899, 990, 999, 1000, 1010	828, 843, 847, 880, 881, 884–
<code>\printindex</code>	41, 394, 644, 899	886, 896, 908, 909, 911, 913,
<code>\printnoidglossaries</code>	196,	914, 917–919, 929–933, 935,
	213, 484, 492, 736, 744, 991, 1000	936, 946, 948, 949, 972, 976,
<code>\printnoidglossary</code>		977, 988, 990, 991, 994, 996,
	196, 199, 213, 213,	997, 999, 1002, 1003, 1009,
	219–221, 228, 483, 484, 486,	1010, 1013, 1015, 1018, 1028, 1031
	492, 497, 498, 502, 735, 736,	
	738, 744, 749, 750, 754, 990,	<code>\protected@write</code>
	991, 993, 1000, 1004, 1005, 1010	71, 73, 180–182, 193,
<code>\printnumbers</code>	40, 394, 644, 899	194, 196, 199, 202, 211, 272,
<code>\printsymbols</code>	40, 393, 644, 899	411, 472, 473, 482, 483, 486,
<code>\ProcessOptions</code>		487, 492, 575, 594, 661, 662,
	327, 334, 537, 589, 790, 842	724, 725, 734, 735, 738, 739,
<code>\ProcessOptionsX</code> ..	42, 394, 644, 900	744, 828, 847, 917, 918, 979,
<code>\protect</code>	54, 118, 166, 248–	980, 989, 991, 993, 994, 999, 1031
	250, 255, 256, 271, 352, 353,	
	362, 363, 368, 401, 441, 513,	<code>\protected@xdef</code>
	514, 519, 523, 526, 528, 553,	16–18, 21, 26, 82, 97,
	559, 560, 564, 651, 691, 692,	100, 102, 103, 207, 215, 339,
	765, 766, 771, 775, 779, 781,	340, 380–382, 385, 416, 429,
	806, 812, 813, 817, 907, 946,	430, 489, 540–542, 629, 630,
	971, 1020, 1021, 1025, 1026, 1031	632, 635, 667, 677, 679–681,
<code>\protected@csedef</code> .	91, 422, 673, 929	741, 745, 794, 795, 884–886,
<code>\protected@csxappto</code>	1031	889, 923, 933, 935–937, 997, 1001
<code>\protected@csxdef</code>		
	91, 422, 673, 929, 1031	<code>\providecommand</code>
<code>\protected@eappto</code>	35,	5,
	74, 75, 101, 203, 641, 662, 663,	20, 34, 44, 45, 52, 71, 110, 135,
	680, 740, 896, 918, 919, 936, 995	182, 193, 196, 216, 276, 297,
<code>\protected@edef</code>		316, 330, 335, 382, 390, 396,
	8, 11, 18, 21, 35, 56, 58,	399, 411, 434, 452, 473, 482,
	61, 63, 64, 67, 71, 72, 74, 75,	484, 486, 494, 504, 505, 538,
	91–97, 100–102, 118, 119, 123,	591, 596, 607, 618, 631, 640,
	125, 126, 166, 175–177, 192,	646, 650, 661, 684, 704, 725,
		734, 736, 738, 746, 756, 757,
		791, 844, 850, 861, 872, 878,
		886, 895, 901, 902, 905, 917,
		941, 957, 980, 989, 991, 1001, 1002

<code>\provideglossaryentry</code>	743, 745, 751, 753, 758, 759,
. 83, 418, 668, 924	788, 793, 794, 806, 807, 820–
<code>\ProvidesFile</code> 45, 396, 646, 902	822, 824, 827, 829–831, 836–
<code>\ProvidesGlossariesLang</code>	838, 847, 850, 857, 861, 864,
. 45, 396, 646, 902	872–878, 881, 882, 884–886,
<code>\ProvidesPackage</code>	892, 893, 896, 898–901, 909,
. 4, 270, 273, 276, 280,	917, 919, 920, 923, 925, 933,
286, 291, 297, 302, 310, 316,	935, 938, 940, 941, 947, 949–
326, 327, 334, 373, 374, 537,	956, 972, 980–983, 985, 986,
572, 573, 578, 589, 594–596,	988, 989, 991, 992, 995, 997–
598, 602, 604, 607, 611, 615,	1001, 1006–1009, 1013, 1028, 1031
618, 624, 790, 825, 826, 832,	<code>\renewacronymstyle</code> . 248, 362–365,
842, 847, 848, 850, 852, 855,	367–370, 512, 559–562, 564–
858, 861, 864, 868, 872, 877, 1031	566, 764, 812–815, 817, 819, 1019
	<code>\renewcommand</code> 5–7, 9–15, 18–20, 22–
	24, 28, 30, 31, 34, 36–38, 40–
	42, 46–49, 62, 75, 77, 79, 94,
	104, 108–110, 175, 177–179,
	185, 186, 193–198, 215, 216,
	226, 227, 241, 245, 246, 248–
	258, 274, 275, 277–285, 287,
	288, 290, 292–307, 310–315,
	317–325, 335, 339, 348, 350,
	352, 353, 360–371, 374–379,
	381–384, 387–394, 396–398,
	406, 412, 414, 424, 432, 433,
	470–472, 476, 477, 482–485,
	493, 494, 502, 511–522, 524,
	525, 527–529, 531–533, 535,
	538, 540, 546, 549, 551–553,
	558–570, 573, 574, 578–588,
	595–613, 615–617, 619–629,
	631–634, 636–640, 642, 643,
	645, 647, 648, 656, 663–665,
	675, 683, 684, 721–723, 728,
	729, 734–737, 745, 746, 753,
	754, 763–778, 780, 782, 784–
	786, 788, 791, 793, 794, 799,
	802, 804, 806, 811–823, 826,
	827, 831–841, 848–867, 869–
	883, 885–888, 891–893, 895,
	897, 899, 900, 902–904, 912,
	919–921, 931, 938, 940, 941,
	976–978, 983, 984, 989–992,
	1001, 1002, 1009, 1016, 1018–1027
	<code>\renewenvironment</code>
	. 230, 274, 277, 280–283, 285,
	286, 288–317, 319, 320, 323,
	504, 595, 597–601, 603–622,
	756, 848, 850, 852–874, 876, 1011
R	
<code>\r</code> 26, 385, 635, 889	
<code>\raggedright</code>	
. 289–296, 310–316, 603–	
607, 615–618, 856–861, 869–872	
<code>\raisebox</code> 134, 451, 703, 957	
<code>\ref</code> 229, 503, 755, 1011	
<code>\refstepcounter</code> 103,	
104, 229, 503, 755, 937, 938, 1010	
<code>\regex</code> 234, 1012	
<code>\relax</code> 7, 10,	
13, 17, 18, 20, 30, 31, 35, 36,	
39, 41, 42, 57, 71, 76, 78, 82,	
85, 97, 100, 105, 109, 110, 121,	
122, 125, 126, 128–133, 167,	
182, 184–186, 188, 189, 192,	
194, 196–198, 203, 207, 208,	
210, 213, 215, 223, 226, 234,	
235, 263, 272, 276, 277, 290,	
297, 301, 302, 316, 318, 319,	
321–325, 338, 339, 353–355,	
375, 377, 378, 380–382, 388,	
389, 391, 393–395, 403, 411,	
413, 414, 417, 418, 426, 428,	
430, 433, 442–451, 466, 474–	
479, 482–485, 488–491, 493,	
499, 501, 502, 506, 507, 535,	
540, 553, 554, 567–570, 574,	
576–578, 583–585, 594, 597,	
603, 607, 610, 611, 618–624,	
627, 628, 630, 631, 638, 641–	
645, 653, 661, 663, 664, 667,	
669, 677, 679, 681, 683, 684,	
693, 695–702, 717, 718, 726–	
728, 730, 731, 733–737, 740–	

<code>\renewglossarystyle</code>	<code>\setbool</code>	28, 387, 637, 891
..... 236 , 507 , 759 , 1013	<code>\setbox</code>	290 , 603 , 604 , 857
<code>\RequireGlossariesLang</code> 45 ,	<code>\setcounter</code>
46 , 326 , 327 , 396 , 397 , 572 ,	... 104 , 229 , 503 , 755 , 937 , 1010	
589 , 646 , 647 , 825 , 842 , 902 , 903	<code>\SetCustomDisplayStyle</code> 263 , 534 , 535 , 787 , 788
<code>\RequirePackage</code> 4–6 , 12 , 13 ,	<code>\SetCustomStyle</code>	263 , 535 , 788
29 , 30 , 42 , 45 , 263 , 280 , 286 ,	<code>\SetDefaultAcronymDisplayStyle</code>	... 260 , 521 , 522 , 773 , 774 , 1028
291 , 297 , 302 , 310 , 326 , 327 ,	<code>\SetDefaultAcronymStyle</code> 260 , 522 , 533 , 774 , 786
335 , 374 , 378 , 388 , 394 , 396 ,	<code>\SetDescriptionAcronymDisplayStyle</code> 261 , 526 , 527 , 778 , 780
535 , 537 , 538 , 572 , 589 , 598 ,	<code>\SetDescriptionAcronymStyle</code> 261 , 527 , 533 , 780 , 787
602 , 604 , 607 , 611 , 615 , 624 ,	<code>\SetDescriptionDUAAcronymDisplayStyle</code> 261 , 524–526 , 777 , 778
627 , 628 , 637 , 638 , 644 , 646 ,	<code>\SetDescriptionDUAAcronymStyle</code> 261 , 525 , 533 , 778 , 787
788 , 790 , 791 , 825 , 842 , 852 ,	<code>\SetDescriptionFootnoteAcronymDisplayStyle</code> 260 , 522 , 524 , 775–777
855 , 858 , 861 , 864 , 868 , 877 ,	<code>\SetDescriptionFootnoteAcronymStyle</code> 261 , 524 , 533 , 776 , 787
878 , 882 , 892 , 893 , 900 , 902 , 1028	<code>\SetDUADisplayStyle</code> 262 , 532 , 533 , 785 , 786
<code>\restorecounters@</code> 125 , 444 , 695 , 949	<code>\SetDUASyle</code> ..	262 , 533 , 534 , 786 , 787
<code>restoremakegloss</code> (option)	<code>\setentrycounter</code> 56 , 182 , 225 , 235 , 402 ,
34 473 , 501 , 506 , 573 , 652 , 725 ,	753 , 758 , 826 , 908 , 980 , 1008 , 1013
<code>\romannumeral</code> . 203 , 206 , 207 , 322 ,	<code>\SetFootnoteAcronymDisplayStyle</code> 261 , 528 , 529 , 780 , 782
324 , 488 , 489 , 585 , 622 , 623 ,	<code>\SetFootnoteAcronymStyle</code> 261 , 529 , 534 , 782 , 787
740 , 741 , 838 , 875 , 876 , 995 , 996	<code>\SetGenericNewAcronym</code>	245 ,
 247 , 511 , 512 , 763 , 764 , 1018 , 1019	
	<code>\setglossarypreamble</code> 50 , 399 , 649 , 904	
	<code>\setglossarysection</code> 52 , 400 , 650 , 905	
	<code>\setglossarystyle</code>	214 ,
 226 , 235 , 263 , 278–289 , 292–	
 302 , 304–309 , 311–313 , 315 ,	
 318 , 320 , 321 , 325 , 493 , 506 ,	
 507 , 535 , 597–603 , 605–621 ,	
 623 , 745 , 758 , 759 , 788 , 850–	
 875 , 877 , 1000 , 1009 , 1013 , 1029	
	<code>\setglossentrycompatibility</code> 232 , 501 , 505 , 507 , 753 , 757 , 759
	<code>\setkeys</code>	29 , 33 , 42 , 52 , 96 ,
 124 , 176 , 177 , 214 , 245 , 387 ,	
 390 , 395 , 400 , 425 , 443 , 470 ,	

471, 493, 511, 522, 524, 525,
527, 529, 531, 533, 535, 637,
640, 645, 650, 676, 694, 722,
745, 763, 774, 776, 778, 780,
782, 784, 786, 788, 892, 895,
901, 905, 932, 948, 977, 1000, 1018
\setlength
. 279, 280, 291, 297–301, 303,
310, 317, 319, 320, 324, 585,
598, 604, 607–609, 611, 615,
619–623, 838, 851, 852, 858,
861–865, 868, 869, 872–874, 876
\SetSmallAcronymDisplayStyle . .
. 262, 530–532, 783–785
\SetSmallAcronymStyle
. 262, 531, 534, 784, 787
\setstylefile . . 47, 398, 648, 903, 904
\settodepth 105, 938
\settoheight 104, 451, 703, 938
\settowidth
105, 249, 514, 584, 585, 622,
623, 766, 838, 875, 876, 938, 1021
\setupglossaries . . 42, 394, 645, 900
\sfcodes 13, 378, 628, 882
short (key) 80
short-long (acrstyle) 249, 362
short-long-desc (acrstyle) 252, 364
shortaccess (key) 337
shortcuts (option) 32
shortplural (key) 80
shortpluralaccess (key) 337
\show 263–
269, 372, 373, 535–537, 571,
572, 788–790, 824, 825, 1029, 1030
\showacronymlists 268, 537, 790, 1030
\showglocounter . 265, 535, 789, 1029
\showglodesc 266, 536, 789, 1029
\showglodescaccess 372, 572, 825
\showglodescplural 266, 536, 789, 1030
\showglodescpluralaccess
. 372, 572, 825
\showglofield 268, 537, 790, 1030
\showglofirst 264, 535, 789, 1029
\showglofirstaccess 372, 571, 824
\showglofirstpl 264, 535, 789, 1029
\showglofirstpluralaccess
. 372, 571, 825
\showgloflag 267, 536, 790, 1030
\showgloindex 267, 536, 790, 1030
\showglolevel 264, 535, 788, 1029
\showgloloclist . . 267, 537, 790, 1030
\showglolong 267, 536, 790, 1030
\showglolongaccess 373, 572, 825
\showglolongpluralaccess
. 373, 572, 825
\showglongname 266, 536, 789, 1029
\showglongnameaccess 372, 571, 824
\showgloparent 263, 535, 788, 1029
\showgloplural 264, 535, 788, 1029
\showglopluralaccess 372, 571, 824
\showgloshort 267, 536, 789, 1030
\showgloshortaccess 373, 572, 825
\showgloshortpluralaccess
. 373, 572, 825
\showglosort 266, 536, 789, 1030
\showglossaries . . 268, 537, 790, 1030
\showglossarycounter
. 269, 537, 790, 1030
\showglossaryentries
. 269, 537, 790, 1030
\showglossaryin . . 268, 537, 790, 1030
\showglossaryout 268, 537, 790, 1030
\showglossarytitle 268, 537, 790, 1030
\showglosymbol 266, 536, 789, 1030
\showglosymbolaccess 372, 571, 825
\showglosymbolplural
. 267, 536, 789, 1030
\showglosymbolpluralaccess
. 372, 571, 825
\showglotext 264, 535, 788, 1029
\showglotextaccess 372, 571, 824
\showglotype 264, 535, 789, 1029
\showglouser 265, 536, 789, 1029
\showglouserii 265, 536, 789, 1029
\showglouseriii 265, 536, 789, 1029
\showglouseriv 265, 536, 789, 1029
\showglouserv 265, 536, 789, 1029
\showglouservi 266, 536, 789, 1029
sm-short-long (acrstyle) 251
sm-short-long-desc (acrstyle) 253, 365
smallcaps (option) 32
smaller (option) 32
\SmallNewAcronymDef 262,
371, 530, 532, 570, 783, 785, 823
sort (key) 76
sort (option) 15
\space 9, 34–36, 39,
44, 55, 56, 60, 63, 71, 78, 80,
108, 111–113, 118, 174, 180–
184, 188, 190–196, 199, 204,

211, 214, 216, 230, 242, 248–	759, 786, 826–831, 847, 880,
258, 262, 273, 275, 277, 278,	885, 888, 890, 895, 896, 898,
281, 292, 303, 310, 311, 317–	901, 907–913, 917, 918, 920–
319, 321, 323, 324, 352, 353,	922, 925–928, 936, 940–943,
362–371, 376, 390–393, 396,	949, 950, 952, 953, 955, 956,
402, 405, 407, 414, 415, 432,	976–983, 985, 986, 989–991,
434, 435, 441, 442, 469, 472–	993, 995, 997, 998, 1000–1002,
475, 479–486, 491, 493, 494,	1004, 1005, 1010, 1016, 1028, 1031
503, 504, 507, 510, 513–521,	<code>\strut</code> 277–279,
526, 530, 552, 553, 559–567,	281, 282, 284, 292, 293, 295,
573–577, 579–587, 595–599,	303, 305, 307, 311, 312, 314,
604, 611, 615, 619–623, 626,	321, 506, 579–582, 584, 586–
640–643, 646, 652, 655, 657,	588, 597–600, 604–606, 611–
664–666, 683, 685, 691–693,	613, 615–617, 621, 758, 833–
721, 724–727, 731–738, 743,	835, 837, 839–841, 850–854,
745, 746, 755, 759, 761, 765–	858–860, 865–867, 869–871, 875
773, 779, 783, 806, 812–820,	<code>style</code> (option) <u>11</u>
826–830, 833, 835–841, 848–	styles:
852, 858, 865, 869, 872–876,	<code>altlist</code> <u>278</u>
880, 895, 896, 898, 901, 908,	<code>altlistgroup</code> <u>278</u>
910–913, 917, 920–922, 940,	<code>altlisthypergroup</code> <u>279</u>
942, 943, 946, 976, 979–981,	<code>altlong4col</code> <u>285</u>
985–991, 993, 995, 999, 1000,	<code>altlong4col-booktabs</code> <u>288</u>
1002, 1011, 1016, 1020–1028, 1032	<code>altlong4colborder</code> <u>286</u>
<code>\spacefactor</code> 13, 378, 628, 882	<code>altlong4colheader</code> <u>285</u>
<code>\SS</code> 27, 386, 636, 890	<code>altlong4colheaderborder</code> <u>286</u>
<code>\ss</code> 27, 386, 636, 890	<code>altlongragged4col</code> <u>295</u>
<code>\str</code> 235, 1013	<code>altlongragged4col-booktabs</code> <u>289</u>
<code>\string</code> 9, 18, 19,	<code>altlongragged4colborder</code> <u>296</u>
24, 27, 34–36, 39, 44, 54–56,	<code>altlongragged4colheader</code> <u>295</u>
58–61, 63, 71, 73, 78, 80, 86,	<code>altlongragged4colheaderborder</code>
89, 90, 101, 102, 108, 110–113, <u>296</u>
126, 127, 129, 130, 133, 174,	<code>altsuper4col</code> <u>308</u>
178–185, 188, 189, 193–197,	<code>altsuper4colborder</code> <u>309</u>
199, 204, 208–210, 214, 216,	<code>altsuper4colheader</code> <u>308</u>
219–221, 228, 241, 262, 272,	<code>altsuper4colheaderborder</code> <u>309</u>
373, 374, 376, 381, 384, 390–	<code>altsuperragged4col</code> <u>314</u>
393, 396, 401–407, 411, 414,	<code>altsuperragged4colborder</code> <u>315</u>
415, 418, 419, 421, 429, 432,	<code>altsuperragged4colheader</code> <u>315</u>
434, 435, 441, 442, 445, 447,	<code>altsuperragged4colheaderborder</code>
448, 450, 469, 471–476, 478, <u>315</u>
479, 482–486, 490, 491, 493,	<code>almtree</code> <u>322</u>
494, 497, 498, 502, 505–507,	<code>almtreegroup</code> <u>325</u>
573–578, 594, 626, 631, 634,	<code>almtreehypergroup</code> <u>325</u>
640–643, 646, 651, 652, 654–	<code>index</code> <u>317</u>
657, 661, 662, 664–666, 669–	<code>indexgroup</code> <u>318</u>
672, 680, 683–685, 692, 693,	<code>indexhypergroup</code> <u>318</u>
696, 698–700, 702, 721–728,	<code>inline</code> <u>273</u>
730, 731, 734–738, 742, 743,	<code>list</code> <u>277</u>
745, 746, 749, 750, 754, 757,	<code>listdotted</code> <u>279</u>

listgroup	277	super4colborder	307
listhypergroup	278	super4colheader	307
long	280	super4colheaderborder	308
long-booktabs	287	superborder	304
long3col	282	superheader	304
long3col-booktabs	287	superheaderborder	304
long3colborder	283	superragged	310
long3colheader	283	superragged3col	312
long3colheaderborder	283	superragged3colborder	313
long4col	283	superragged3colheader	313
long4col-booktabs	288	superraggedborder	311
long4colborder	284	superraggedheader	311
long4colheader	284	superraggedheaderborder	312
long4colheaderborder	285	superraggedright3colheaderborder	313
longborder	281	tree	319
longheader	281	treegroup	320
longheaderborder	282	treehypergroup	320
longragged	291	treenoname	320
longragged-booktabs	289	treenonamegroup	321
longragged3col	293	treenonamehypergroup	321
longragged3col-booktabs	289	subentrycounter (option)	15
longragged3colborder	294	\subglossentry	101, 215, 223, 233, 275, 277–279, 281, 282, 284, 292, 293, 295, 303, 305, 307, 311, 312, 314, 318, 319, 321, 323, 429, 493, 499, 500, 505, 506, 596–600, 604–606, 611– 613, 615–617, 619–622, 680, 745, 751, 757, 849–854, 858– 860, 865–867, 869–871, 873, 874, 876, 936, 1001, 1007, 1012
longragged3colheader	294	\subitem	297, 317, 318, 583, 607, 618, 619, 836, 861, 872, 873
longragged3colheaderborder	294	sublistdotted (style)	280
longraggedborder	292	\subsubitem	297, 317, 318, 583, 607, 619, 836, 861, 872, 873
longraggedheader	292	super (style)	303
longraggedheaderborder	293	super3col (style)	305
mcolalttree	301	super3colborder (style)	305
mcolalttreegroup	301	super3colheader (style)	306
mcolalttreehypergroup	301	super3colheaderborder (style)	306
mcolalttreespannav	302	super4col (style)	306
mcolindex	297	super4colborder (style)	307
mcolindexgroup	297	super4colheader (style)	307
mcolindexhypergroup	298	super4colheaderborder (style)	308
mcolindexspannav	298	superborder (style)	304
mcoltree	298	superheader (style)	304
mcoltreegroup	299	superheaderborder (style)	304
mcoltreehypergroup	299	superragged (style)	310
mcoltreename	300		
mcoltreenamegroup	300		
mcoltreenamehypergroup	300		
mcoltreenamespannav	300		
mcoltreespannav	299		
sublistdotted	280		
super	303		
super3col	305		
super3colborder	305		
super3colheader	306		
super3colheaderborder	306		
super4col	306		

superragged3col (style)	312	\texttt	239, 509, 761, 1015
superragged3colborder (style) . . .	313	\textulc	241, 509, 761, 1016
superragged3colheader (style) . . .	313	\textup 239, 241, 509, 761, 1015, 1016	
superraggedborder (style)	311	\TH	27, 386, 636, 890
superraggedheader (style)	311	\th	27, 386, 636, 890
superraggedheaderborder (style) .	312	\the 43, 46, 56, 61, 63, 72, 128–134,	
superraggedright3colheaderborder		180, 184, 186, 188, 200, 202,	
(style)	313	206, 207, 211, 223, 238, 246,	
symbol (key)	76	248–250, 254, 255, 362, 366,	
symbolaccess (key)	336	368, 371, 395, 397, 402, 405,	
\symbolname . 44 , 47 , 284 , 285 , 288 ,		407, 411, 446–451, 472, 475,	
296 , 307–309 , 315 , 316 , 396 ,		477, 478, 486, 487, 489, 492,	
397 , 601 , 602 , 607 , 613–615 ,		499, 504, 505, 508, 511–514,	
618 , 646 , 647 , 854 , 856 , 860 ,		518, 519, 521–532, 534, 538,	
861 , 867 , 868 , 871 , 872 , 901 , 903		559, 560, 563, 564, 567–571,	
symbolplural (key)	77	573, 574, 577, 646, 647, 652,	
symbolpluralaccess (key)	336	656, 657, 661, 697–703, 724,	
symbols (option)	40	727–730, 738, 739, 741, 744,	
		751, 756, 757, 760, 763–766,	
		769, 771, 773–777, 779, 781,	
		783–785, 787, 788, 791, 812,	
		813, 816, 817, 820–824, 826,	
		828, 830, 901, 902, 908, 911,	
		913, 917, 951–956, 978, 981,	
		983, 985, 993, 994, 996, 999,	
		1007, 1015, 1018–1021, 1024, 1025	
		\the@numberlist 175, 470, 721, 722, 976	
		\theglossary	6, 374, 624, 878
		theglossary (env.)	230
		\theglossaryentry	14,
		230, 379, 503, 628, 755, 883, 1011	
		\theglossarysubentry	15,
		230, 379, 504, 629, 755, 883, 1011	
		\theglentrycounter . . .	125, 204,
		207, 444, 488, 489, 574, 695,	
		740, 741, 827, 948, 949, 996, 997	
		\theH	209, 490, 742, 998
		\theHglossaryentry .	14, 379, 628, 883
		\theHglossarysubentry	
		15, 379, 629, 883
		\theHglentrycounter	
		125, 204, 207, 444, 445, 488,	
		489, 695, 740, 741, 949, 996, 997	
		\thesection	43, 395, 645, 901
		\this@dialect	46, 326, 327,
		397, 572, 589, 647, 825, 842, 903	
		\tiny	8, 376, 625, 879
		\tl	5, 878
		toc (option)	10

`\toks@` [43](#), [46](#), [56](#), [61](#), [63](#), [72](#), [128–134](#),
[186](#), [188](#), [211](#), [238](#), [395–397](#),
[402](#), [405](#), [407](#), [411](#), [446–451](#),
[476–478](#), [492](#), [504](#), [505](#), [508](#),
[538](#), [573](#), [646](#), [647](#), [652](#), [656](#),
[657](#), [661](#), [697–703](#), [728–730](#),
[744](#), [756](#), [757](#), [760](#), [791](#), [826](#),
[901](#), [902](#), [908](#), [911](#), [913](#), [917](#),
[951–956](#), [983](#), [985](#), [999](#), [1014](#), [1015](#)
`\toprule` [287](#), [288](#), [602](#), [855](#), [856](#)
`\TrackIfKnownLanguage` [45](#), [902](#)
`\trans@languages` [46](#), [396](#), [646](#), [902](#)
`\translate` [46](#), [47](#), [397](#), [647](#), [648](#), [902](#), [903](#)
`translate` (option) [30](#)
`\translatelet` [19](#), [20](#), [40](#), [381](#), [382](#), [393](#),
[394](#), [631](#), [632](#), [644](#), [885](#), [886](#), [899](#)
`tree` (style) [319](#)
`treegroup` (style) [320](#)
`treehypergroup` (style) [320](#)
`treenoname` (style) [320](#)
`treenonamegroup` (style) [321](#)
`treenonamehypergroup` (style) [321](#)
`\triangleright` [8](#), [376](#), [625](#), [879](#)
`\ttfamily` [8](#), [376](#), [625](#), [879](#)
`\TX@trial` [105](#), [430](#), [681](#), [938](#)
`type` (key) [77](#)
`\typeout` [23](#), [383](#), [633](#), [887](#)

U

`\u` [26](#), [385](#), [635](#), [889](#)
`\uccode` [223](#), [499](#), [751](#), [1006](#)
`ucmark` (option) [14](#)
`\undef` [79](#), [85](#), [212](#), [414](#), [418](#),
[492](#), [665](#), [669](#), [744](#), [921](#), [925](#), [999](#)
`\unexpanded` [41](#), [900](#)
`\unskip` [94](#), [279](#),
[424](#), [580](#), [598](#), [675](#), [833](#), [851](#), [931](#)
`\unvbox` [290](#), [603](#), [604](#), [857](#)
`\usedictionary` [46](#), [396](#), [647](#), [902](#)
`\usepackage` [27](#), [219–](#)
[221](#), [241](#), [373](#), [374](#), [497](#), [498](#),
[749](#), [750](#), [890](#), [1004](#), [1005](#), [1016](#)
`user1` (key) [79](#)
`user1access` (key) [337](#)
`user2` (key) [79](#)
`user2access` (key) [337](#)
`user3` (key) [79](#)
`user3access` (key) [337](#)
`user4` (key) [79](#)
`user4access` (key) [338](#)
`user5` (key) [79](#)
`user5access` (key) [338](#)
`user6` (key) [80](#)
`user6access` (key) [338](#)
`\UTFviii@two@octets` [27](#), [386](#), [636](#), [890](#)
`\UTFviii@two@octets@combine` [27](#), [386](#), [636](#), [890](#)

V

`\v` [26](#), [386](#), [635](#), [890](#)
`\value` [104](#), [937](#)
`\vbox` [290](#), [603](#), [604](#), [857](#)
`\vsize` [290](#), [291](#), [604](#), [857](#)
`\vskip` [276](#), [290](#), [297](#), [316](#), [597](#),
[603](#), [607](#), [618](#), [850](#), [857](#), [861](#), [872](#)
`\vss` [290](#), [603](#), [604](#), [857](#)

W

`\warn@nomakeglossaries` [192](#), [194](#), [195](#),
[197](#), [482–484](#), [733–736](#), [988–991](#)
`\warn@noprintglossary` [194–196](#), [212](#),
[215](#), [482–484](#), [492](#), [493](#), [734–](#)
[736](#), [744](#), [745](#), [989–991](#), [999](#), [1001](#)
`\write` [35](#), [86](#), [110](#), [180–185](#),
[190–193](#), [196](#), [200](#), [216](#), [391](#),
[418](#), [419](#), [434](#), [472–476](#), [480](#),
[481](#), [484](#), [487](#), [494](#), [574–578](#),
[641](#), [669](#), [684](#), [685](#), [724–728](#),
[732](#), [733](#), [736](#), [738](#), [746](#), [828–](#)
[831](#), [896](#), [925](#), [941](#), [942](#), [978–](#)
[982](#), [986–989](#), [991](#), [994](#), [1001](#), [1002](#)
`writelabelnames` (option) [36](#)
`writelabels` (option) [36](#)
`\writeist` [179](#),
[189](#), [472](#), [475](#), [476](#), [479](#), [574](#),
[577](#), [578](#), [724](#), [726](#), [728](#), [731](#),
[828](#), [830](#), [831](#), [978](#), [981](#), [982](#), [986](#)

X

`\x` [508](#)
`\xatlevel@` [125](#), [444](#), [695](#), [948](#)
`\xcapitalisewords` [467](#)
`\xdef` [98](#), [426](#),
[428](#), [493](#), [594](#), [677](#), [847](#), [848](#), [933](#)
`\xglaccsupp` [344](#), [344](#), [543](#), [544](#), [797](#)
`\xglfieldaccsupp` [343](#), [543](#), [796](#)
`\xifinlist` [84](#), [668](#), [924](#)

<code>\xifinlistcs</code>	217, 218, 221, 495, 498, 746, 747, 750, 1002, 1003, 1005	<code>\xspace</code>	240, 509, 761, 1015
<code>xindy</code> (option)	33	Y	
<code>xindygloss</code> (option)	34	<code>\year</code>	180, 184, 472, 475, 574, 577, 724, 727, 828, 830, 978, 981
<code>xindynoglsnumbers</code> (option)	34	Z	
<code>\xmakefirstuc</code>	<code>\Z</code>	234, 1012
 435–437, 441, 465, 466, 590, 686–688, 692, 717, 718, 843	<code>\z@</code>	290, 603, 604, 857